

---

-----  
README FOR NVIDIA CUDA Visual Profiler  
Version 1.0  
-----

Published by  
NVIDIA Corporation  
2701 San Tomas Expressway  
Santa Clara, CA 95050

#### Notice

BY DOWNLOADING THIS FILE, USER AGREES TO THE FOLLOWING:

ALL NVIDIA SOFTWARE, 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 NON-INFRINGEMENT, 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 or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. These materials supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

#### Trademarks

NVIDIA, CUDA, and the NVIDIA logo are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

#### Copyright

(C) 2007-2008 by NVIDIA Corporation. All rights reserved.

#### LIST OF SUPPORTED FEATURES:

-----  
- Execute a CUDA program with profiling enabled and view the profiler output as a table. The table has the following columns for each GPU method:  
    timestamp: Start time stamp  
    method: GPU method name. This is either "memcpy" for memory copies or the name of a GPU kernel.  
GPU Time  
CPU Time  
Occupancy  
Profiler counters:  
    gld\_incoherent : Number of non-coalesced global memory loads  
    gld\_coherent : Number of coalesced global memory loads  
    gst\_incoherent : Number of non-coalesced global memory stores  
    gst\_coherent : Number of coalesced global memory stores  
    local\_load : Number of local memory loads  
    local\_store : Number of local memory stores  
    branch : Number of branch events (instruction and/or sync stack)  
    divergent\_branch : Number of divergent branches within a warp  
    instructions : Number of dynamic instructions (in fetch)  
    warp\_serialize : Number of threads in a warp serialize based on address (GRF or constant)  
    cta\_launched : Number of CTAs launched on the PM TPC

Please refer the "Interpreting Profiler Counters" section below for more information on profiler counters.

Note that profiler counters are also referred to as profiler signals.

- Display the summary profiler table. It has the following columns for each GPU method:
  - method name
  - number of calls
  - total GPU time
  - total CPU time
  - % age GPU time
  - Total counts for each profiler counter.
- Display various kinds of plots:
  - . Summary profiling data bar plot
  - . GPU Time Height plot
  - . GPU Time Width plot
  - . Profiler counter bar plot
  - . Profiler output table column bar plot
- Analysis of profiler output - lists out method with high number of:
  - . incoherent stores
  - . incoherent loads
  - . warp serializations
- Compare profiler output for multiple program runs of the same program or for different programs. Each program run is referred to as a session.
- Save profiling data for multiple sessions. A group of sessions is referred to as a project.
- Import/Export CUDA Profiler CSV format data

#### DESCRIPTION OF DIFFERENT PLOTS:

- 
- Summary profiling data bar plot
    - . One bar for each method
    - . Bars sorted in decreasing gpu time,
    - . Bar length is proportional to cumulative gputime for a method
  - GPU Time Height Plot:
 

It is a bar diagram in which the height of each bar is proportional to the GPU time for a method and a different bar color is assigned for each method. A legend is displayed which shows the color assignment for different methods. The width of each bar is fixed and the bars are displayed in the order in which the methods are executed.

When the "fit in window" option is enabled the display is adjusted so as to fit all the bars in the displayed window width. In this case bars for multiple methods can overlap. The overlapped bars are displayed in decreasing order of height so that all the different bars are visible.

When the "Show CPU Time" option is enabled the CPU time is shown as a bar in a different color on top of the GPU time bar. The height of this bar is proportional to the difference of CPU time and GPU time for the method.
  - GPU Time Width Plot:
 

It is a bar diagram in which the width of each bar is proportional to the GPU time for a method and a different bar color is assigned for each method. A legend is displayed which shows the color assignment for different methods. The bars are displayed in the order in which the methods are executed. When time stamps are enabled the bars are positioned based on the time stamp.

The height of each bar is based on the option chosen:

    - a) fixed height : height is fixed.
    - b) height proportional to instruction issue rate: the instruction issue rate for a method is equal to profiler "instructions" counter value divided by the gpu time for the method.
    - c) height proportional to incoherent load + store rate: the incoherent load + store rate for a method is equal to the sum of profiler

---

"gld\_incoherent" and "gst\_incoherent" counter values divided by the gpu time for the method.

- Profiler counter bar plot  
It is a bar plot for profiler counter values for a method from the profiler output table or the summary table.
  - . One bar for each profiler counter
  - . Bars sorted in decreasing profiler counter value
  - . Bar length is proportional to profiler counter value
- Profiler output table column bar plot  
It is a bar plot for any column of values from the profiler output table or summary table
  - . One bar for each row in the table
  - . Bars sorted in decreasing column value
  - . Bar length is proportional to column value

#### STEPS FOR SAMPLE cudaprof USAGE:

##### SAMPLE1:

- Open a new project using main menu option File->New or toolbar  
Select the project name and project directory where the project files will be saved.
- Select the session settings through the dialog  
Browse and select the CUDA program to profile.  
Change the working directory if it is different from the program directory.  
Select option for profiler counters  
Select option for time stamps  
Change maximum program execution time (if needed)
- Execute the CUDA program by clicking the Start button of the Session settings dialog or through the main menu option "Profile->Start"  
If the CUDA program is correctly executed the profiler output will be displayed.
- To display the summary table right click on "Session1" in the session list. Choose the "Summary table" option. Or use the "Summary table" tool bar option.
- To display the GPU Time summary plot right click on "Session1" in the session list and choose the "GPU Time Summary Plot" option. Or use the "GPU Time Summary Plot" tool bar option.
- You can scroll, resize or reposition the profiler output and GPU Time Summary plot windows.
- Save the project by using the main menu option "File->Save" or the toolbar.
- Exit cudaprof using the main menu option "File->Exit".

##### SAMPLE2:

- Open the project saved in SAMPLE1 or one of the sample projects using the main menu option "File->Open". The profiler output table will be displayed.
- To display the GPU Time Height plot right click on "Session1" in the session list. Choose the "GPU Time Height Plot" option. Also try the "GPU Time Width Plot".
- Select settings for a new session by using the main menu option "Profile->Session settings".  
Browse and select the CUDA program to profile.  
Change the working directory if it is different from the program directory.
- Execute the CUDA program by clicking the Start button of the Session settings dialog or through the main menu option "Profile->Start"  
If the CUDA program is correctly executed the profiler output will be displayed.

- Compare the profiler output for "Session1" and "Session2".
- Try the "Profiler counter plot" and "Column plot" by right clicking on the appropriate row or column in the profiler output or summary table for a session.
- Exit cudaprof using the main menu option "File->Exit".

#### BRIEF DESCRIPTION OF SOME cudaprof GUI COMPONENTS:

- 
- Top line shows the main menu options: File, Profile, Session, Options, Window and Help. See the description below for details on the menu options.
  - Second line has 4 groups of tool bar icons.
    - File tool bar group has:
      - New project
      - Open existing project and
      - Save project
    - Profile tool bar group has:
      - Session settings
      - Start profiling
    - Session tool bar group has:
      - Summary table
      - Summary plot
      - GPU time height plot
      - GPU time width plot
    - View options tool bar group has:
      - Session view settings
  - Left vertical window lists all the sessions in the current project
    - Right clicking on a session brings up the context sensitive menu. See the description below for details on the menu options.
  - Right workspace area contains windows which include:
    - . Tabbed window for each session. The different windows for a session are shown as different tabs:
      - . Profiler output table
      - . Summary table
      - . GPU Time height plot
      - . GPU Time width plot
      - . Profiler counter plot
      - . Column plot
  - Output window - displays standard output & standard error for the CUDA program which is run. Also some additional status messages are displayed in this window.

#### MAIN MENU

##### "File" menu

- |
- New : Create a new project
  - The "New project" dialog is opened to choose the project name and project directory. On OK the "Session settings" dialog is opened.
- |
- Open : Open an existing project
  - The "Open project" dialog is opened to select the profiler project to be opened. On "Open" the project data for all sessions is loaded and the profiler data table is displayed.
- |
- Save : Save the current project
  - The profiler data for the current open project is saved to the disk.
- |

---

```
--- Save As : Save the current project as a new project.
              The project name & directory can be selected.
              The profiler data for the current open project is
              saved to the disk.
|
--- Close : Close the current project
              The current open project is closed. All profiler session data
              is deleted from memory and all open windows are closed.
|
--- Import: Import CUDA profiler output in comma separated format (CSV).
              A new session is created in the current project and
              imported data is loaded.
|
--- Export: Export CUDA profiler output for the current session to a
              file in the comma separated format (CSV).
|
--- List of recently opened profiler projects.
|
--- Exit   : Exit the cudaprof program
```

#### "Profile" menu

```
|
--- Session settings : Change session settings
|
--- Start           : Start CUDA program with profiling enabled
```

#### "Session" menu

```
|
--- Copy settings to current: Copy settings for the current session
                              as the session settings to be used for
                              a new profiling session.
|
--- View: Various profiler data viewing options for the current session.
|
--- Analyze profiler counters: Analyze profiler counters values for the
                              current session.
                              This is same as the profiler table context menu
                              "Analyze profiler counters" option.
|
--- Delete: Delete the current session.
              This is same as the Session context menu "Delete" option.
|
--- Properties: Show the properties for the current session.
              This is same as the Session context menu "Properties" option.
|
--- Rename: Rename the current session.
```

#### "Session->View" menu

```
|
--- Summary Table: View summary profiler table for current session.
                  The summary table has the following columns:
                  - method name
                  - number of calls
                  - total GPU time
                  - total CPU time
                  - % age GPU time
                  - Cumulative counters count columns for each available
                    profiler counters
                  The rows in the table are sorted in decreasing order of total GPU
                  time and memcopy is shown as the last row.
|
--- GPU Time Summary plot : View GPU time summary plot for current session.
                              This is same as the Session context menu "GPU Time
                              Summary plot" option.
|
--- GPU Time Height plot  : View GPU time height plot for current session.
```

---

This is same as the Session context menu "GPU Time Height plot" option.

|  
 --- GPU Time Width plot : View GPU time width plot for current session.  
 This is same as the Session context menu "GPU Time Width plot" option.

#### "Options" menu

|  
 --- Session view settings: Change session view settings for the current session  
 |  
 --- Default view settings: Change the default view settings to be used for new sessions  
 |  
 --- Height plot: Change global GPU time height plot options.  
 |  
 --- Plot Colors: Select colors for plots.  
 |  
 --- Show output window: Enable / disable display of output window.  
 |  
 --- Session window layout settings: Change settings for display of multiple session windows.  
 |  
 --- Environment variable settings: Change environment variable settings used by the CUDA program.  
 |  
 --- Demangle method names: Enable / disable method name de-mangling.

#### "Options->Height Plot" menu

|  
 --- Use Global Scale: Enable / disable option to use a common global scale across multiple sessions.

#### "Options->Plot Colors" menu

|  
 --- Method Colors: Pop ups a color dialog which can be used to select colors used for different methods in plots. The colors are saved on application exit and so they can be used across cudaprof sessions.

#### "Window" menu

|  
 --- Close: Close active window  
 |  
 --- Close All: Close all open windows  
 |  
 --- Tile: Tile all open windows  
 |  
 --- Cascade: Cascade all open windows

#### "Help" menu

|  
 --- About: Display cudaprof program version and copyright information.

#### TOOL BARS

##### File tool bar group:

--- Create a new project: The behaviour is same as the "File->New" menu option  
 --- Open an existing project: The behaviour is same as the "File->Open" menu option  
 --- Save the current project: The behaviour is same as the "File->Save" menu option

##### Profile tool bar group:

--- Session settings: The behaviour is same as the "Profile->Session settings" menu option

---

--- Start profiling: The behaviour is same as the "Profile->Start" menu option

Session tool bar group:

- Summary table: The behaviour is same as the "Session->View->Summary table" menu option
- Summary plot: The behaviour is same as the "Session->View->Summary plot" menu option
- GPU time height plot: The behaviour is same as the "Session->View->GPU time height plot" menu option
- GPU time width plot: The behaviour is same as the "Session->View->GPU time width plot" menu option

View options tool bar group has:

- Session view settings: The behaviour is same as the "Options->Session View Settings" menu option

DIALOGS

"New project" dialog

- Project Name: Name of the profiler project
  
- Project location: Directory where the project files will be saved

"Session settings" dialog

"Session" Tab

- Session Name: Name of the profiler session  
By default a new session name is chosen ("Session1", "Session2", ...).  
This can be changed by the user.
  
- Launch: Select the CUDA program to be profiled.
  
- Working directory: Select the working directory to be used for running the CUDA program.
  
- Arguments: Command line arguments to be passed to the CUDA program.
  
- Max. execution time (in seconds): Select maximum time to wait for CUDA program execution completion. After this cutoff time the program is aborted.
  
- Run in separate window: This option is useful for console applications which accept some keyboard input. In this case the CUDA program is run from a separate window. The standard output and standard error for the CUDA program is shown in this separate window.  
Note that currently this option is supported only on Linux and a new "xterm" window is opened.

"Configuration" Tab

- Enable time stamp: Enable option to include time stamps for methods.  
This feature is available only with CUDA version 1.1 or later.
  
- Counter List: You can select or de-select all counters by using the "Counter List" check box. You can also select any sub-set of specific counters using the check boxes for each counters.  
Since a maximum of only 4 profiler counters can be enabled for a single run - the CUDA program is run multiple times if more than 4 counters are selected.  
This feature is available only with CUDA version 1.1 or later.

"Session View Settings" dialog

This dialog can be invoked using the main menu option

"Options->Session View Settings" or the toolbar. This dialog allows changing settings for the different views for the current session. There is a separate tab for different views. The dialog is opened with the tab corresponding to the current view. Only tabs for currently created views can be selected.

---

"Profiler Table" Tab

--- Hide All Zero Counters: Enable / disable hiding of counter columns having all zero values.  
This is enabled by default.

--- Columns Shown: Lists columns which are to be shown.  
Can select & move columns from hidden list to shown list using "<<".

--- Columns Hidden: Lists columns which are to be hidden.  
Can select & move columns from shown list to hidden list using ">>".

"Summary Table" Tab

--- Show Average Data: Enable / disable showing average data values.  
When this option is disabled the sum total across all the calls for a method are shown.  
When this option is enabled the total value is divided by the number of times the method is called and this average value for a method is displayed.  
This option is disabled by default.

--- Columns Shown: Lists columns which are to be shown.  
Can select & move columns from hidden list to shown list using "<<".

--- Columns Hidden: Lists columns which are to be hidden.  
Can select & move columns from shown list to hidden list using ">>".  
The CPU usec column is hidden by default.

"Summary Plot" Tab

--- Percentage Displayed: Enable / disable displaying percentage values.  
When this option is disabled total values are shown.  
This option is enabled by default.

--- Average Displayed: Enable / disable using average data values.  
When this option is disabled total values are used.  
This option is disabled by default.

"Height Plot" Tab

--- Show legend: Enable / disable display of GPU Time plot legend  
--- Fit in window: Enable / disable option to fit the GPU plot in the window.  
When fit is enabled multiple bars can overlap.  
--- Show CPU Time: Enable / disable option to show CPU time.

"Width Plot" Tab

--- Enable Time Stamp: Enable / disable option to use time stamps.  
--- Show CPU Time: Enable / disable option to show CPU time.  
--- Fit in window: Enable / disable option to fit the plot in the window.  
--- Max Width of Bar: Maximum width of a bar in pixels. For this option the plot display is immediately updated & so one can interactively choose an appropriate value.  
--- Bar Height Option: Choose option to use for bar height.

"Default View Settings" dialog

This dialog can be invoked using the main menu option "Options->Default View Settings". This dialog allows changing the default settings which are used subsequently for new session views which are displayed. The description of settings is same as those for the "Session View Settings" dialog.

"Method Colors" dialog

This dialog is invoked using the main menu option "Options->Plot Colors->Method Colors". This dialog allows user to select the colors which are used for different methods in plots. These colors are saved on cudaprof exit and can be used across cudaprof sessions.

SESSION LIST CONTEXT MENU

|

```

--- Copy settings to current: Same as menu option
                                "Session->Copy settings to current"
|
--- Summary table: Display the profiler summary table.
|
--- GPU Time Summary Plot: Display the GPU Time Summary plot for the
                                selected session. The GPU time summary plot
                                options can be changed using the main menu
                                option "Options->GPU Time Summary Plot".
|
--- GPU Time Height Plot: Display the GPU Time Height plot for the
                                selected session. The GPU time Height plot options
                                can be changed using the "Session View Settings"
                                dialog.
|
--- GPU Time Width Plot: Display the GPU Time Width plot for the selected
                                session. The GPU time width plot options can be
                                changed using the "Session View Settings" dialog.
|
--- Delete: Delete the selected session
|
--- Properties: Show the project and session settings for the
                                selected session.
|
--- Rename: Rename the selected session.

```

#### PROFILER TABLE CONTEXT MENU

```

|
--- Profiler counter plot: Display the profiler counter plot for the
                                method in the current row.
|
--- Column plot: Display the column plot for the current column.
|
--- Analyze profiler counters: Analyze profiler counter values. This option is
                                enabled only for the summary table. This highlights any methods
                                which have a high rate of incoherent loads or a high rate of
                                incoherent stores or a high rate of warp serialization. These
                                rates are calculated as the cumulative profiler counter count value
                                divided by the cumulative gpu time for a method.
|
--- Export: Export the profiler data to a CSV format file.
|
--- Copy: Copy the selected table cells to the clipboard.
|
--- Average data: Show average data values instead of totals in the
                                summary table.

```

#### cuda prof PROJECT FILES SAVED TO DISK

```

-----
<project-name>.cpj                : Cuda profiler project file
<project-name>_<session-name>.csn : Cuda profiler session file
<project-name>_<session-name>.csv : Cuda profiler session data file

```

#### cuda prof SETTINGS WHICH ARE SAVED

```

-----
Following is the list of cuda prof settings which are saved and remembered
across different cuda prof sessions.

```

```

Last opened project path
Method Colors
Recent files list
Recent programs
Recent work Dirs
Show Output window
Demangle Method Names

```

```

Main Window/Size
Main Window/Maximized
Global view dialog/Size

```

---

Session view dialog/Size  
Horizontal Splitter/Sizes  
Vertical Splitter/Sizes

Profiler Table/Hide Zero Columns  
Summary Table/Show Average  
Summary Plot/Average Displayed  
Summary Plot/Percentage Displayed  
Height Plot/Fit in window  
Height Plot/Show CPU Time  
Height Plot/Show Legend  
Height Plot/Use global scale  
Width Plot/Enable time stamp  
Width Plot/Fit in window  
Width Plot/Maximum bar width  
Width Plot/Show CPU Time  
Width Plot/Show legend  
Width Plot/Start time stamp at zero  
Width Plot/Type

On Windows these settings are saved in the system registry  
at the location "HKEY\_CURRENT\_USER\Software\NVIDIA\cuda\_prof".

On Linux these settings are saved to the file  
"\$HOME/.config/NVIDIA/cuda\_prof.conf".

#### Interpreting Profiler Counters

-----

The performance counter values do not correspond to individual thread activity. Instead, these values represent events within a thread warp. For example, a divergent branch within a thread warp will increment the `divergent_branch` counter by one. So the final counter value stores information for all divergent branches in all warps.

In addition, the profiler can only target one of the multiprocessors in the GPU, so the counter values will not correspond to the total number of warps launched for a particular kernel. For this reason, when using the performance counter options in the profiler the user should always launch enough threads blocks to ensure that the target multiprocessor is given a consistent percentage of the total work. In practice, it is best to launch at least around 100 blocks for consistent results.

For the reasons listed above, users should not expect the counter values to match the numbers one would get by inspecting kernel code. The values are best used to identify relative performance differences between unoptimized and optimized code. For example, if for the initial version of the program the profiler reports `N` non-coalesced global loads, it is easy to see if the optimized code produces less than `N` non-coalesced loads. In most cases, the goal is to make `N` go to 0, so the counter value is useful for tracking progress toward this goal.