DLProf Plugin for TensorBoard

User Guide

DU-09833-002 _v21.04 | April 2021

Table of Contents

Chapter 1. DLProf TensorBoard Plugin....1
1.1. Overview....1
1.2. What's New in 1.2.0....1
1.3. Features....1
Chapter 2. Quick Start....3
2.1. Installing Using Python Wheel....3
2.2. Using the NGC Docker Container....3
2.3. Generating TensorBoard Event Files....4
2.4. Starting TensorBoard....4
Chapter 3. DLProf Plugin....5
3.1. Overview....5
3.2. Pane Overview....5
3.3. Navigation Pane....6
3.4. Details Pane....8
3.4.1. Op Details Panel....8
3.4.2. Kernel Details Panel....9
Chapter 4. Content Pane....10
4.1. Dashboard....10
4.1.1. GPU Utilization Panel....10
4.1.2. Op Summary Panel....11
4.1.3. Kernel Summary Panel....12
4.1.4. Tensor Core Kernel Efficiency Panel....13
4.1.5. Performance Summary Panel....13
4.1.6. Iteration Summary panel....15
4.1.7. Top 10 GPU Ops Panel....15
4.1.8. System Configuration Panel....16
4.1.9. Recommendations Panel....17
4.1.10. Guidance Panel....18
4.2. Op Type Summary....18
4.3. Ops and Kernels....20
4.3.1. Ops Data Table....21
4.3.2. Kernel Summaries Data Table....22
4.4. Kernels by Iteration....23
4.4.1. Iteration Summary Data Table....23
4.4.2. Kernels in Selected Iteration....24

4.5. Kernels by Op 25
4.5.1. Iteration Summary Data Table 26
4.5.2. Ops in Selected Iteration Table 27
4.5.3. Kernels Selected Iteration / Op Combination Table 28
4.6. Iterations view 28
4.7. GPUs View 32
Chapter 5. Using Data Tables 34
Chapter 6. Troubleshooting FAQ 36

Chapter 1. DLProf TensorBoard Plugin

1.1. Overview

The DLProf Plugin for TensorBoard makes it easy to visualize the performance of your models by showing Top 10 operations that took the most time, eligibility of Tensor Core operations and Tensor Core usage, as well as interactive iteration reports.

1.2. What's New in 1.2.0

- Compatible with event files generated by DLProf v1.1.0 / r21.04.
- Ability to select and switch between previous aggregations
- Removed support for the DLProf version of the GRAPHS plugin.

1.3. Features

This release includes these commands and features:

- Panelized Dashboard Summary View: A summary view comprising several panels that provide a quick overview of the performance results.
- Top-level Key Metrics: The Summary view displays several key metrics that are used to quickly gauge the quality of the performance, including Average Iteration Time and Tensor Core Utilization.
- Top 10 GPU Ops Node: A table in the Summary view lists the top 10 Op Nodes with the most time spent on the GPU.
- Expert Systems Panel: This panel displays any issues detected by the DLProf Expert Systems, along with suggestions on how to address the issues and improve the models performance.
- Iteration Summary Panel: This panel visually displays iterations. Users can quickly see how many iterations are in the model, the iterations that were aggregated/profiled, and the durations of tensor core kernels in each iteration.

- Interactive Tables: All tables in detailed views are completely interactive, allowing the use to sort, filter, and paginate the display.
- Interoperable Tables: Several views have the ability to drill down for more information. Selecting a row in one table will populate the next table with performance information pertaining to the selection.
- Client/server architecture:
  - All of the data is now in a relational database.
  - Load times have improved for all views.
- GPUs View: Displays GPU utilization and GPU properties of all GPUs used during profiling.

Chapter 2. Quick Start

2.1. Installing Using Python Wheel

A modified version of TensorBoard 1.15 and the DLPROF Plugin for TensorBoard can be installed from the NVIDIA PY index.

Before installing, note that all previous versions of tensorboard must first be uninstalled prior to installing the DLPROF Plugin for TensorBoard. The plugin requires an NVIDIA modified version of TensorBoard. Use the following command to uninstall Tensorboard if needed.

```
$ pip uninstall -y tensorboard
```

Next, install `nvidia-pyindex`.

```
$ pip install nvidia-pyindex
```

Then, install Nvidia's version of Tensorboard.

```
 $ pip install nvidia-tensorboard
```

Finally install the DLProf Plugin for TensorBoard:

```
$ pip install nvidia-tensorboard-plugin-dlprof
```

2.2. Using the NGC Docker Container

Make sure you log into NGC as described in Prerequisites before attempting the steps in this section. Use docker pull to get the TensorFlow container from NGC:

```
$ docker pull nvcr.io/nvidia/tensorflow:<21.02>-tf1-py3
```

Where `<xx.yy>` is the version of the TensorFlow container that you want to pull.

Assuming the training data for the model is available in /full/path/to/training/data, you can launch the container with the following command:

```
$ docker run --rm --gpus=1 --shm-size=1g --ulimit memlock=-1 \
--ulimit stack=67108864 -it -p6006:6006 -v/full/path/to/training/data:/data \
nvcr.io/nvidia/tensorflow:<xx.yy>-tf1-py3
```

2.3. Generating TensorBoard Event Files

Event files are created directly from the Deep Learning Profiler. Refer to the Deep Learning Profiler User Guide for information on how to generate the appropriate TensorBoard Event Files.

2.4. Starting TensorBoard

TensorBoard and the GPU Plugin are installed in the TensorFlow 1.x and TensorFlow 2.x container on the NVIDIA GPU Cloud (NGC). The container must be run with the -p6006:6006 option to open port 6006 for the TensorBoard server. (Note, use any port number such as 6007, 6008, etc.)

TensorBoard is launched directly from the container:

```
tensorboard --logdir <event_files>
```

Where `<event_files>` is the path to the event files directory. Once running, TensorBoard can be viewed in a browser with the URL:

```
http://<machine IP Address>:6006
```

Chapter 3. DLProf Plugin

This section describes each of the available views in the DLProf Plugin.

3.1. Overview

The following information is common to all views within the DLProf Plugin.

Terms and Definitions

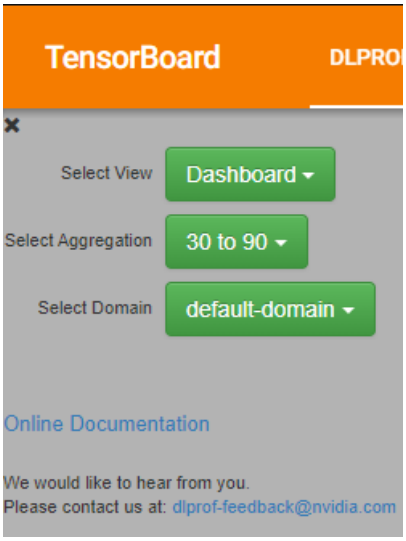
Term	Definition
Aggregation	The rollup of metrics, given an iteration start, iteration stop, and key node.
Op Node	A node in the graph where an operation is performed on the incoming tensor.
Model, Graph, Network	<synonyms>

3.2. Pane Overview

The DLProf plugin user interface is divided into three panes:

Pane	Definintion
Navigation Pane	This pane is where you can navigate to different views and select different NVTX ranges. Additionally, you can navigate to the online documentation and launch your email app to send us comments. Click the 'X' to remove the navigator Pane.
Content Pane	Otherwise known as the VIEWS pane, this is where you will see all of the different pieces of the profiled neural network.
Details Pane	This pane is to display additional details about the information selected in the content pane. Let's say you're looking at the Kernel Summary panel in the content pane, and you want to see more information about the kernels. Click on the  button (tooltip = 'Show Kernel Details'). The detailed kernel information will appear in the Details Pane. Click the 'X' button to remove the details pane.

3.3. Navigation Pane



UI Controls

Control	Definition
X	Closes the Navigator pane and widens all the other panels.
Select View	Clicking on the green "Select View" button provides a drop-down list of available views. Clicking on a name in the drop down list loads that view in the main display panel. Available views are: ▶ Dashboard ▶ Op Type Summary ▶ Ops and Kernels ▶ Kernels by Iteration ▶ Kernels by Op ▶ Iterations ▶ GPUs (when more than one GPU exists in profile run)
Select Domain (optional)	This optional drop down list appears when a network has been profiled using the NVIDIA Tools Extension (NVTX) plugin. See github for more details: https://github.com/NVIDIA/nvtx-plugins Each domain is profiled independently, so when a different domain is selected, all values in the entire DLProf plugin will change. This plugin allows network programmers to isolate and profile certain areas of a network.
Select Aggregation (optional)	This optional drop down list appears when a network has been re-aggregated (ie, aggregated more than once). An aggregation is a combination of iteration start, iteration stop, and key node. The iter stop and iter stop values are listed in the drop list as seen above. To see the corresponding key node, click the drop list and hover over each aggregation in the list. Aggregations can be created in the Iterations view. The workflow and user interface controls are documented in the view.

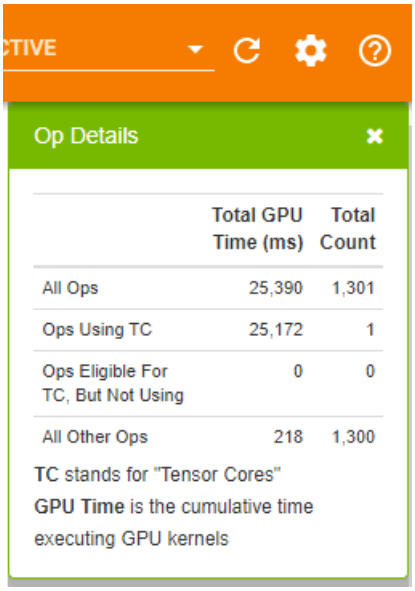
Online Documentation	This hyperlink navigates users to the online version of this document.
Email Us	Let us know! If you have a comment, question, or suggestion, click this link. It will launch your default email software with the TO address already filled in. Just fill in the Subject line, type your email message, and click send.

3.4. Details Pane

The Details pane holds panels that show more details about a particular area of the system.

3.4.1. Op Details Panel

The Op Details panel is displayed at the top-right of the browser underneath the TensorBoard Title Bar.



Fields

Field	Definition
All Ops	Aggregates the total GPU time and count for all ops in the network.
Ops Using TC	Aggregates the total GPU time and count for ops in the network that use Tensor Cores.
Ops Eligible For TC, But Not Using	Aggregates the total GPU time and count for ops in the network that are not using Tensor Cores but are eligible to do so.
All Other Ops	Aggregates the total GPU time and counts for all other types of ops in the network.

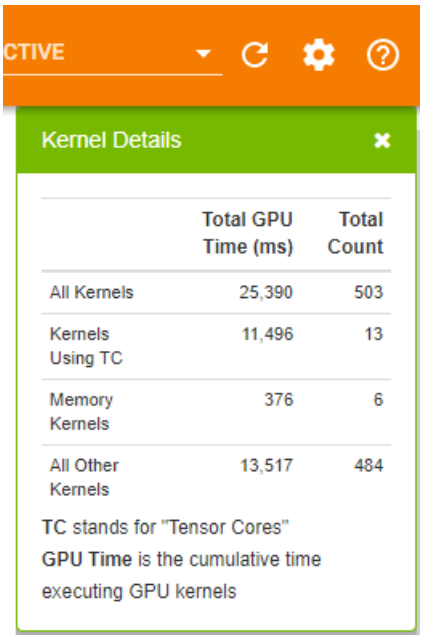
UI Controls

Control	Definition

X	Closes the Details Pane

3.4.2. Kernel Details Panel

The Kernel Details panel is displayed at the top-right of the browser underneath the TensorBoard Title Bar. This panel provides key metrics about the kernels in the network aggregated over the specific iteration range.



Fields

Field	Definition
All Kernels	Aggregates the total GPU time and count for all kernels in the network.
Kernels Using TC	Aggregates the total GPU time and count for all kernels using Tensor Cores.
Memory Kernels	Aggregates the total GPU time and count for all memory kernels.
All Other Kernels	Aggregates the total GPU time and count for all remaining kernel types.

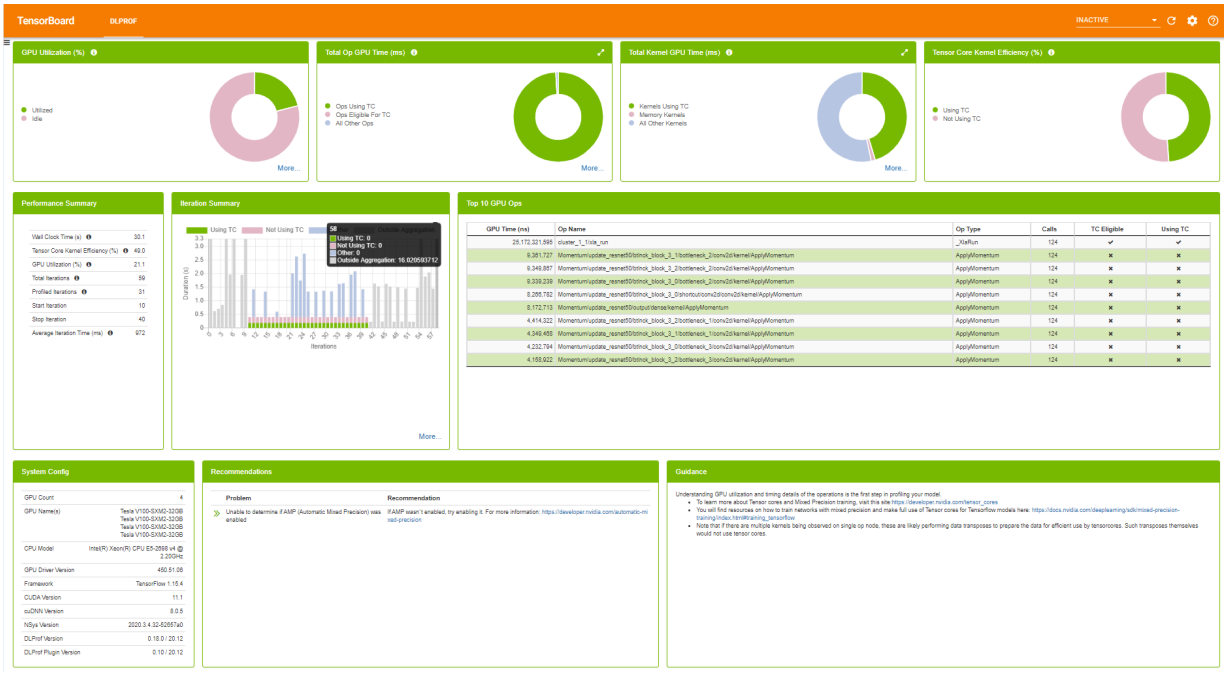
UI Controls

Control	Definition
X	Closes the Details Pane.

Chapter 4. Content Pane

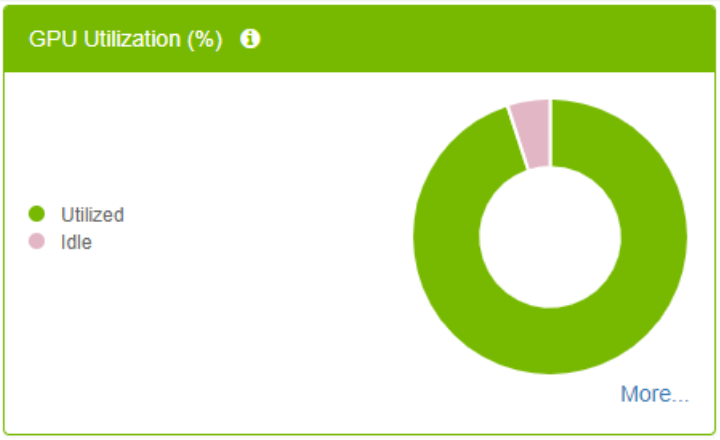
4.1. Dashboard

The Dashboard view provides a high level summary of the performance results in a panelized view. This view serves as a starting point in analyzing the results and provides several key metrics.



4.1.1. GPU Utilization Panel

The GPU Idle panel visually indicates the percentage of GPU utilization time during execution of aggregated iterations. Hovering over a slice in the chart will show the numeric percentage.



Fields

Legend Label	Definition
Utilized	The average GPU utilization percentage across all GPUs.
Idle	The average GPU idle percentage across of all GPUs.

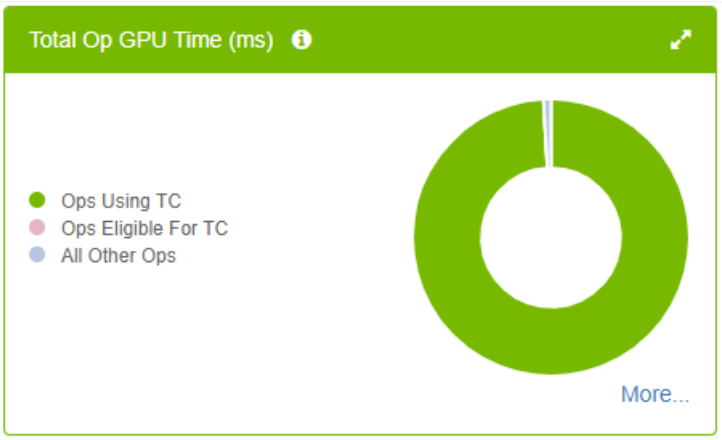
UI Controls

Control	Definition
Legend Entry	Toggle between hiding and showing legend entry in chart.
More...	Show drop-down menu of more views (only visible when more than one GPU was used during profiling).

4.1.2. Op Summary Panel

This panel provides key metrics about the ops in the network aggregated over the specific iteration range. The charts provide a graphical representation of the data.

- Hovering over a slice in the chart will show the aggregated GPU time.
- Clicking a legend item will toggle its visualization in the chart.



Fields

Legend Label	Definition
Ops Using TC	Aggregates the total GPU time for ops in the network that use Tensor Cores.
Ops Eligible For TC	Aggregates the total GPU time for ops in the network that are not using Tensor Cores but are eligible to do so.
All Other Ops	Aggregates the total GPU time for all other types of ops.

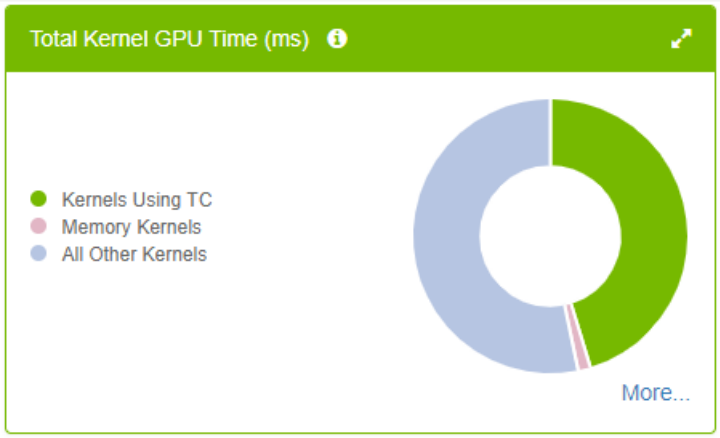
UI Controls

Control	Definition
	Show Op Details panel in Details Pane.
Legend Entry	Toggle between hiding and showing legend entry in chart.
More	Show drop-down menu of more views.

4.1.3. Kernel Summary Panel

The panel provides key metrics about the kernels in the network aggregated over the specific iteration range.

- Hovering over a slice in the chart will show the aggregated GPU time.
- Clicking a legend item will toggle its visualization in the chart.



Fields

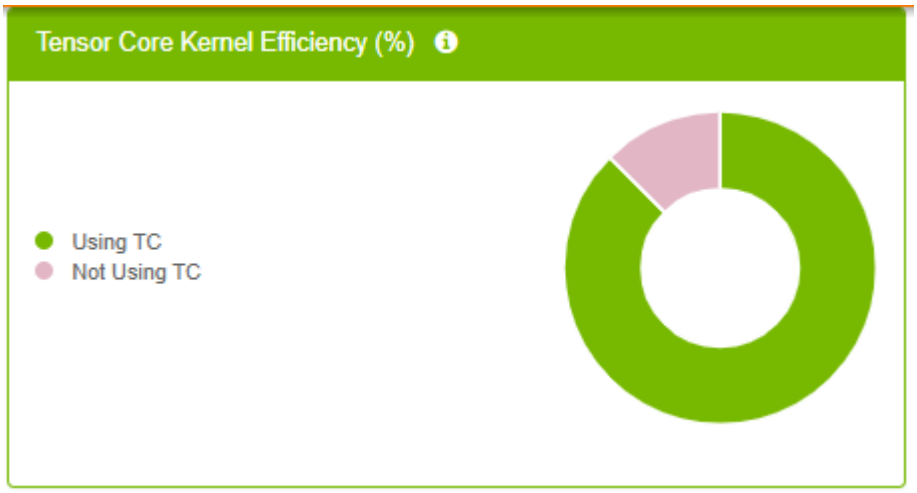
Legend Label	Definition
Kernels Using TC	Aggregates the total GPU time for all kernels using Tensor Cores.
Memory Kernels	Aggregates the total GPU time for all memory-related kernels.
All Other Kernels	Aggregates the total GPU time for all remaining kernel types.

UI Controls

Control	Definition
	Show Kernel Details panel in Details Pane.
More...	Show drop-down menu of more views.

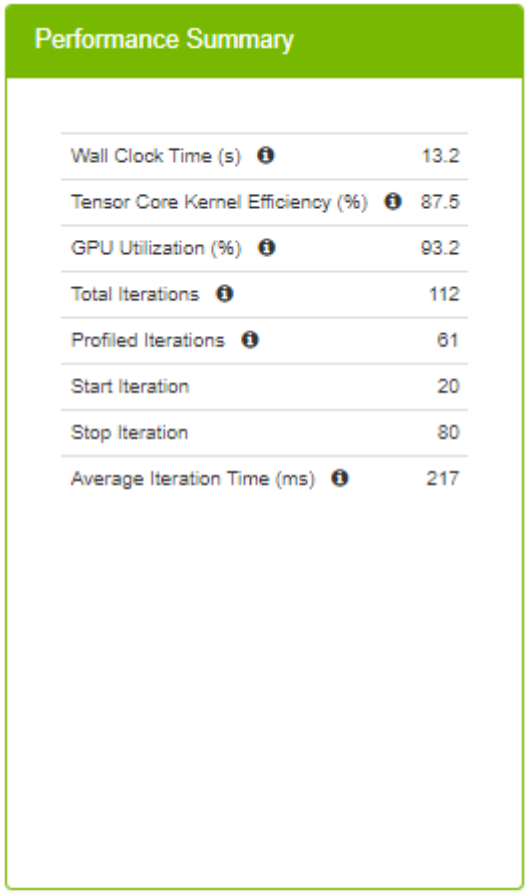
4.1.4. Tensor Core Kernel Efficiency Panel

- Hovering over a slice in the chart will show the percentage.
- Clicking a legend item will toggle its visualization in the chart.



4.1.5. Performance Summary Panel

The Performance Summary panel provides top level key metrics about the performance data aggregated over the specific iteration range. A helpful tooltip text will appear when hovering over the 'i' icon.

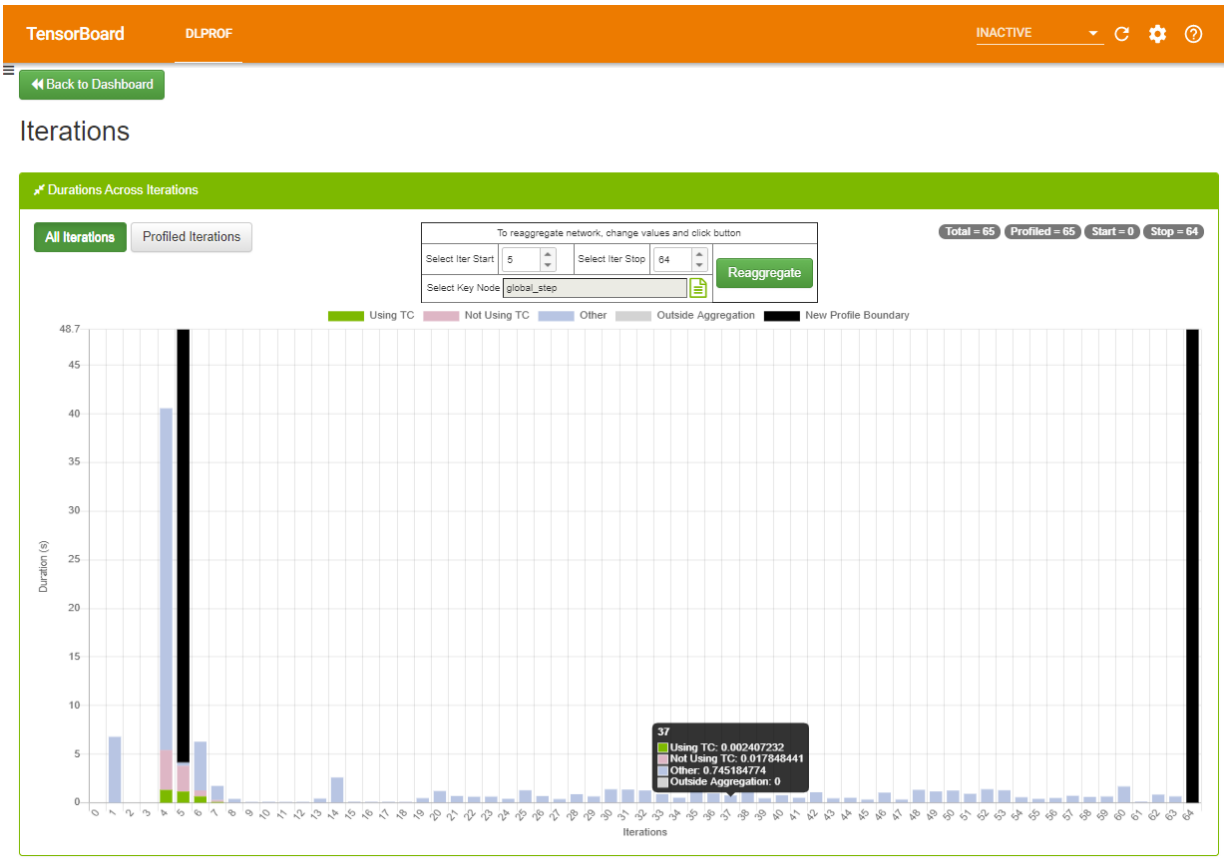


Field	Definition
Wall Clock Time	This is the total run time for the aggregation range, and is defined as the time between the start time of the first op in the starting iteration on the CPU and the end time last op in the final iteration on either the CPU or GPU, whichever timestamp is greatest.
Tensor Core Kernel Efficiency %	This high level metric represents the utilization of Tensor Core enabled kernels. Tensor Core operations can provide a performance improvement and should be used when possible. This metric is calculated by: [Total GPU Time for Tensor Core kernels] / [Total GPU Time for Tensor Core Eligible Ops] A 100% Tensor Core Utilization means that all eligible Ops are running only Tensor Core enabled kernels on the GPU. A 50% Tensor Core Utilization can mean anything from all eligible Ops are running Tensor Core kernels only half of the time to only half of all eligible Ops are running Tensor Core kernels only. This metric should be used with the Op Summary Panel to determine the quality of Tensor Core usage. Higher is better.
GPU Utilization %	Average GPU utilization across all GPUs. Higher is better.

Total Iterations	The total number of iterations found in the network.
Profiled Iterations	The total number of iterations used to aggregate the performance results. This number is calculated using 'Start Iteration' and 'Stop Iteration'.
Start Iteration	The starting iteration number used to generate performance results.
Stop Iteration	The ending iteration number used to generate performance results.
Average Iteration Time	The average iteration time is the total Wall Time divided by the number of iterations.

4.1.6. Iteration Summary panel

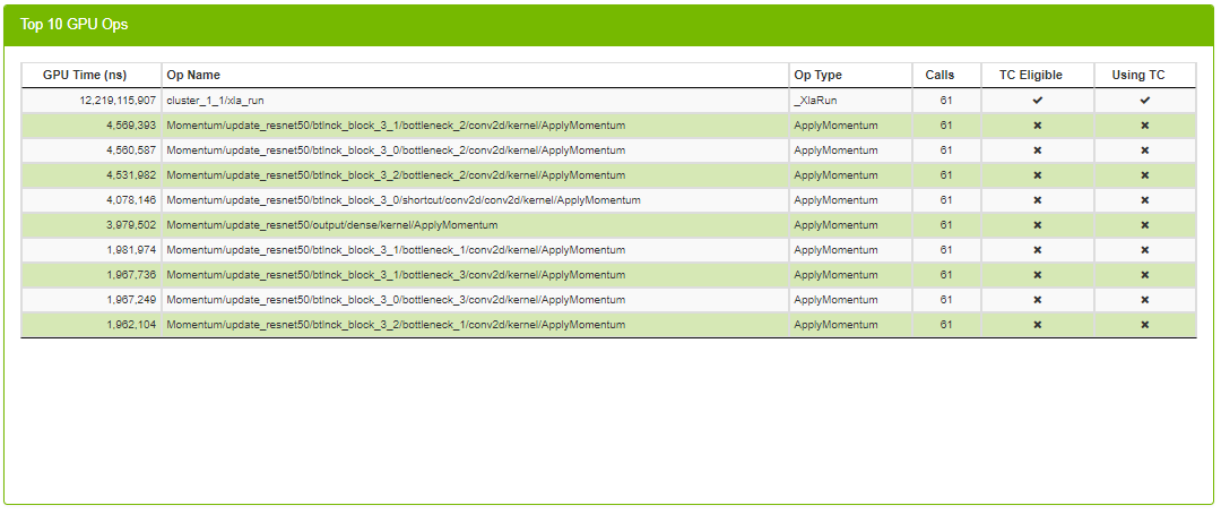
This panel visually displays iterations. Users can quickly see how many iterations are in the model, the iterations that were aggregated/profiled, and the durations of tensor core kernels in each iteration. The colors on this panel match the colors on all the other dashboard panels.



For more information on this panel, see Iterations View.

4.1.7. Top 10 GPU Ops Panel

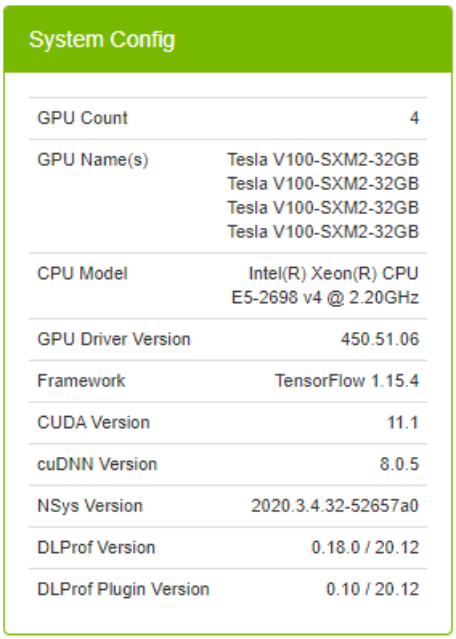
Top 10 GPU Ops table shows the top 10 operations with the largest execution times on the GPU. This table comes pre-sorted with the order of each row in descending GPU Time. The table is not sortable or searchable.


Top 10 GPU Ops

GPU Time (ns)	Op Name	Op Type	Calls	TC Eligible	Using TC
12,219,115,907	cluster_1_1/xla_run	_XlaRun	61	✔	✔
4,569,393	Momentum/update_resnet50/btlnck_block_3_1/bottleneck_2/conv2d/kernel/ApplyMomentum	ApplyMomentum	61	✖	✖
4,560,587	Momentum/update_resnet50/btlnck_block_3_0/bottleneck_2/conv2d/kernel/ApplyMomentum	ApplyMomentum	61	✖	✖
4,531,982	Momentum/update_resnet50/btlnck_block_3_2/bottleneck_2/conv2d/kernel/ApplyMomentum	ApplyMomentum	61	✖	✖
4,078,146	Momentum/update_resnet50/btlnck_block_3_0/shortcut/conv2d/conv2d/kernel/ApplyMomentum	ApplyMomentum	61	✖	✖
3,979,502	Momentum/update_resnet50/output/dense/kernel/ApplyMomentum	ApplyMomentum	61	✖	✖
1,981,974	Momentum/update_resnet50/btlnck_block_3_1/bottleneck_1/conv2d/kernel/ApplyMomentum	ApplyMomentum	61	✖	✖
1,967,736	Momentum/update_resnet50/btlnck_block_3_1/bottleneck_3/conv2d/kernel/ApplyMomentum	ApplyMomentum	61	✖	✖
1,967,249	Momentum/update_resnet50/btlnck_block_3_0/bottleneck_3/conv2d/kernel/ApplyMomentum	ApplyMomentum	61	✖	✖
1,962,104	Momentum/update_resnet50/btlnck_block_3_2/bottleneck_1/conv2d/kernel/ApplyMomentum	ApplyMomentum	61	✖	✖

Column	Definition
GPU Time	Shows total GPU time of all kernels across all GPUs.
Op Name	The name of the op.
Direction	The fprop/bprop direction of the op. (only visible on PyTorch runs).
Op Type	The type of the op.
Calls	The number of times the op was called.
TC Eligible	A true/false field indicating whether or not the op is eligible to use Tensor Core kernels.
Using TC	A true/false field indicating whether or not one of the kernels launched in this op is using Tensor Cores.

4.1.8. System Configuration Panel


System Config

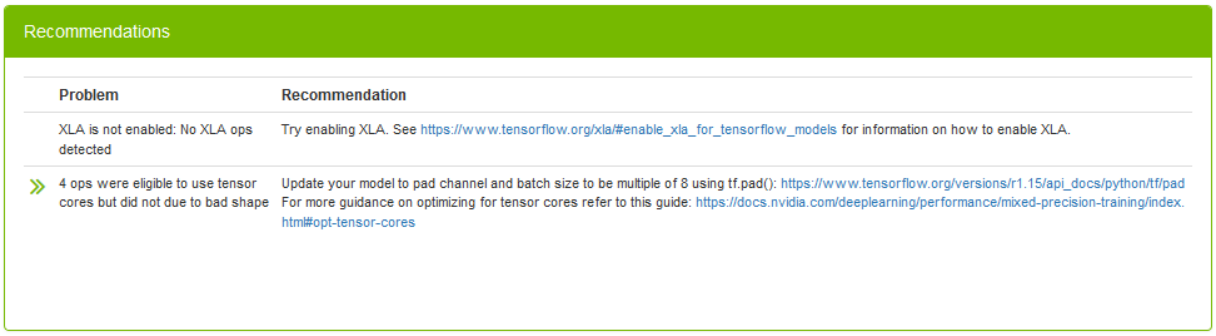
GPU Count	4
GPU Name(s)	Tesla V100-SXM2-32GB Tesla V100-SXM2-32GB Tesla V100-SXM2-32GB Tesla V100-SXM2-32GB
CPU Model	Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz
GPU Driver Version	450.51.06
Framework	TensorFlow 1.15.4
CUDA Version	11.1
cuDNN Version	8.0.5
NSys Version	2020.3.4.32-52657a0
DLProf Version	0.18.0 / 20.12
DLProf Plugin Version	0.10 / 20.12

Field	Definition
Profile Name	(Optional) Helpful label to describe the profiled network. The value in this field corresponds to the value supplied in the --profile_name command line argument in DLProf.
GPU Count	The number of GPU devices found on the computer during training.
GPU Name(s)	A list of the GPU devices found on the computer during training.
CPU Model	The model of the CPU on the computer during training.
GPU Driver Version	The version of the driver used for NVIDIA Graphics GPU.
Framework	The framework used to generate profiling data (eg, TensorFlow, PyTorch).
CUDA Version	The version of the CUDA parallel computing platform.
cuDNN Version	The version of CUDA Deep Neural Network used during training.
NSys Version	The version of Nsight Systems used during training.
DLProf Version	The version of the Deep Learning Profiler used to generate the data visualized in the DLProf Plugin in TensorBoard.
DLProf Plugin Version	The version of the DLProf Plugin inside TensorBoard.
TensorBoard Version	The version of TensorBoard framework.

4.1.9. Recommendations Panel

The Recommendations panel displays common issues detected in the profiled network and provides potential solutions and suggestions to address the issues. The panel will only show issues that have been detected by DLProf. For a full list of potential issues that DLProf looks for, see the Expert Systems section in the Deep Learning Profiler User Guide.

The double-green arrows will show additional information about the detected problem.



Column	Definition
Problem	The description of the scenario that DLProf detected when profiling the network.
>>	(Optional) When present, clicking on the double arrows will display a new view displaying the problem in detail.
Recommendation	A recommendation or actionable feedback, a tangible suggestion that the user can do to improve the network. Clicking on a hyperlink inside the recommendation will open a new tab in the browser.

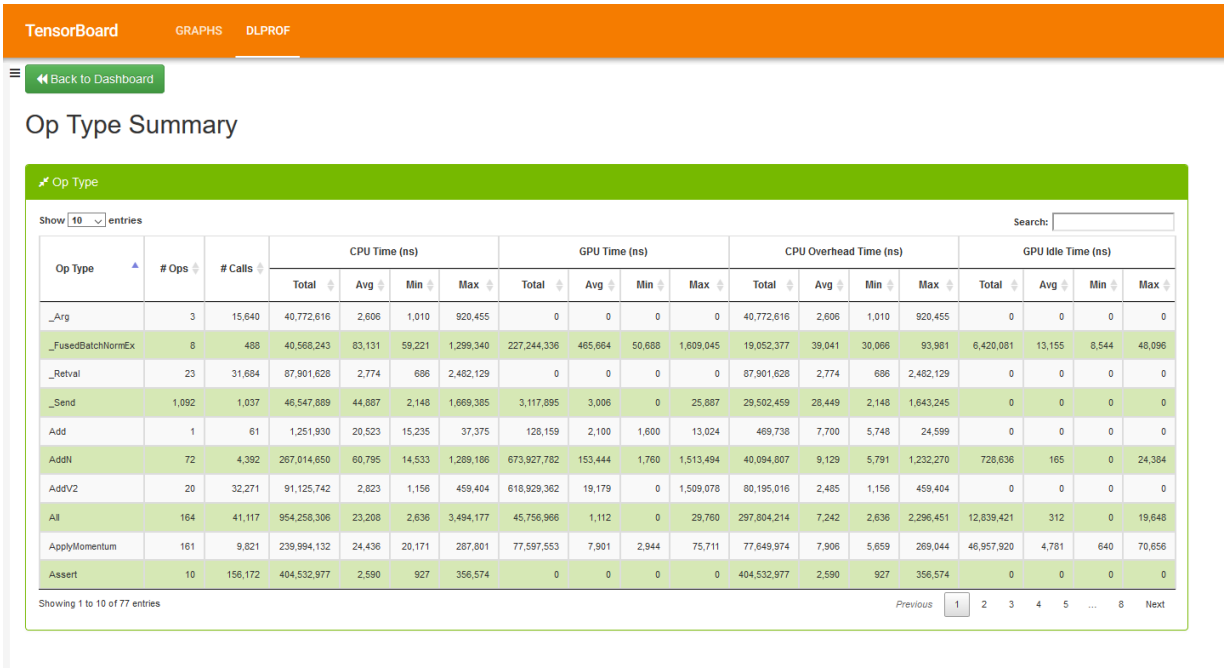
4.1.10. Guidance Panel

This panel provides static guidance to the user to help the user learn more about Tensor Cores, Mixed Precision training. The panel has hyperlinks for further reading. Clicking on a hyperlink inside the Guidance Panel will open a new tab in the browser.

4.2. Op Type Summary

This table aggregates metrics over all op types and enables users to see the performance of all the ops in terms of its types, such as Convolutions, Matrix Multiplications, etc.

See this description for all the features available in all DataTables.



Op Type Data Table

Column name	Description
Op Type	The operation type.
No. Ops	The number of ops that have the Op Type above.
No. Calls	Number of instances that the operation was called / executed.
Total CPU Time (ns)	The total CPU time of all instances of this op type.
Avg. CPU Time (ns)	The average CPU time of all instances of this op type.
Min CPU Time (ns)	The minimum CPU time found amongst all instances of this op type.
Max CPU Time (ns)	The maximum CPU time found amongst all instances of this op type.
Total GPU Time (ns)	The total GPU time of all instances of this op type.
Avg. GPU Time (ns)	The average GPU time of all instances of this op type.
Min GPU Time (ns)	The minimum GPU time found amongst all instances of this op type.

Column name	Description
Max GPU Time (ns)	The maximum GPU time found amongst all instances of this op type.
Total CPU Overhead Time (ns)	The total CPU overhead of all instances of this op type.
Avg. CPU Overhead Time (ns)	The average CPU overhead of all instances of this op type.
Min CPU Overhead Time (ns)	The minimum CPU overhead found amongst all instances of this op type.
Max CPU Overhead Time (ns)	The maximum CPU overhead found amongst all instances of this op type.
Total GPU Idle Time (ns)	The total GPU idle time of all instances of this op type.
Avg. GPU Idle Time (ns)	The average GPU idle time of all instances of this op type.
Min GPU Idle Time (ns)	The minimum GPU idle time found amongst all instances of this op type.
Max GPU Idle Time (ns)	The maximum GPU idle time found amongst all instances of this op type.

4.3. Ops and Kernels

This view enables users to view, search, sort all ops and their corresponding kernels in the entire network.

4.3.1. Ops Data Table

When a row is selected in the Ops table, a summary of each kernel for that op is displayed in the bottom table.

See this description for all the features available in all DataTables.

Entry	Description
GPU Time (ns)	Cumulative time executing all GPU kernels launched for the op.
CPU Time (ns)	Cumulative time executing all op instances on the CPU.
Op Name	The name of the op.
Direction	The fprop/bprop direction of the op. (only visible on PyTorch runs).
Op Type	The operation of the op.
Calls	The number of times the op was called.
TC Eligible	A true/false field indicating whether or not the op is eligible to use Tensor Core kernels. To filter, type '1' for true, and '0' for false.
Using TC	A true/false field indicating whether or not one of the kernels launched in this op is using

Entry	Description
	Tensor Cores. To filter, type '1' for true, and '0' for false.
Kernel Calls	The number of kernels launched in the op.
Data Type	The data type of this op (eg, float16, int64, int32)
Input Shapes	The input shapes of the op. If the contents of this cell is more than 100 characters, a 'See More' hyperlink appears. When clicked, the full contents of the cell appears. When the cell is expanded, the hyperlink text is changed to 'See Less'. When clicked, the cell collapses back to the first 100 characters.
Stack Trace	The stack trace of the op. (only visible on PyTorch runs). If the contents of this cell is more than 100 characters, a 'See More' hyperlink appears. When clicked, the full contents of the cell appears. When the cell is expanded, the hyperlink text is changed to 'See Less'. When clicked, the cell collapses back to the first 100 characters.

4.3.2. Kernel Summaries Data Table

See this description for all the features available in all DataTables.

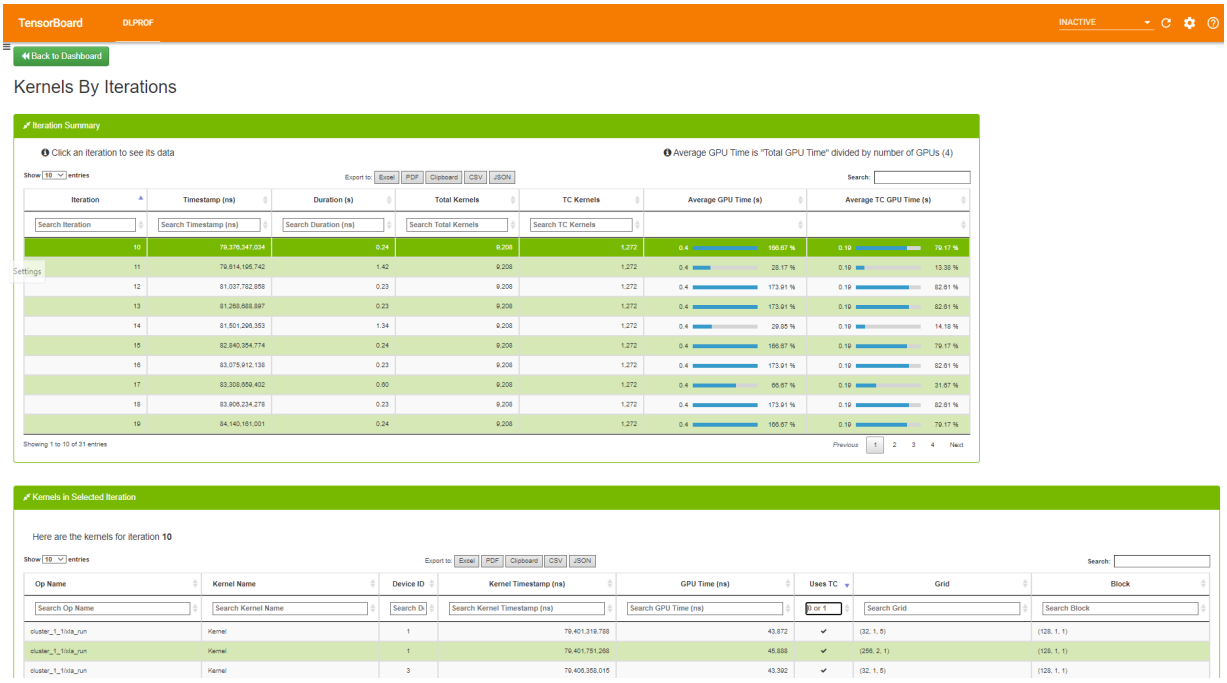
Kernels Summary table

Entry	Description
Kernel name	The full name of the kernel.
Using TC	A true/false field indicating whether or not the kernel is actually using a Tensor Core. To filter, type '1' for true, and '0' for false.
Calls	The number of times this kernel was launched.
GPU Time (ns)	The aggregate duration of each time the kernel was launched.
Avg (ns)	The average duration of each time this kernel was launched.
Min (ns)	The minimum duration of all kernel launches.

Entry	Description
Max (ns)	The maximum duration of all kernel launches.

4.4. Kernels by Iteration

The Kernels by Iterations view shows operations and their execution time for every iteration. At a glance, you can compare iterations with respect to time as well as Kernels executed.



4.4.1. Iteration Summary Data Table

To see the kernels for a specific iteration, click a row in the top table. The 'Kernels in Selected Iteration' table will be filled with the kernels from the selection iteration.

See this description for all the features available in all DataTables.

Column	Description
Iteration	The iteration interval numbers.
Timestamp	The exact time when this iteration started.
Duration	The length of time it took for this iteration to execute. Units on this column are dynamic.
Total Kernels	The number of GPU kernels called during this iteration.

Column	Description
TC Kernels	The number of GPU Tensor Core kernels called during the iteration.
Average GPU Time (µs)	Average execution time for all kernels across all GPUs during the iteration. The contents of this cell are broken into multiple pieces: 1. The value is the average execution time of this iteration across all GPUs. 2. The progress meter visually indicates how much of this iteration executed on the GPU. 3. The percentage is the average GPU time of all kernels across all GPUs over the total iteration time. 4. Hovering over this cell displays a tooltip text with helpful information. Units on this column are dynamic.
Average TC GPU Time (µs)	Average execution time for all Tensor Core kernels across all GPUs during the iteration. The contents of this cell are broken into multiple pieces: 1. The value is the average execution time of Tensor Core kernels during this iteration across all GPUs. 2. The progress meter visually indicates how much of this iteration executed Tensor Core kernels. 3. The percentage is the average GPU time of kernels using Tensor Core across all GPUs over the total iteration time. 4. Hovering over this cell displays a tooltip text with helpful information. Units on this column are dynamic.

4.4.2. Kernels in Selected Iteration

See this description for all the features available in all DataTables.

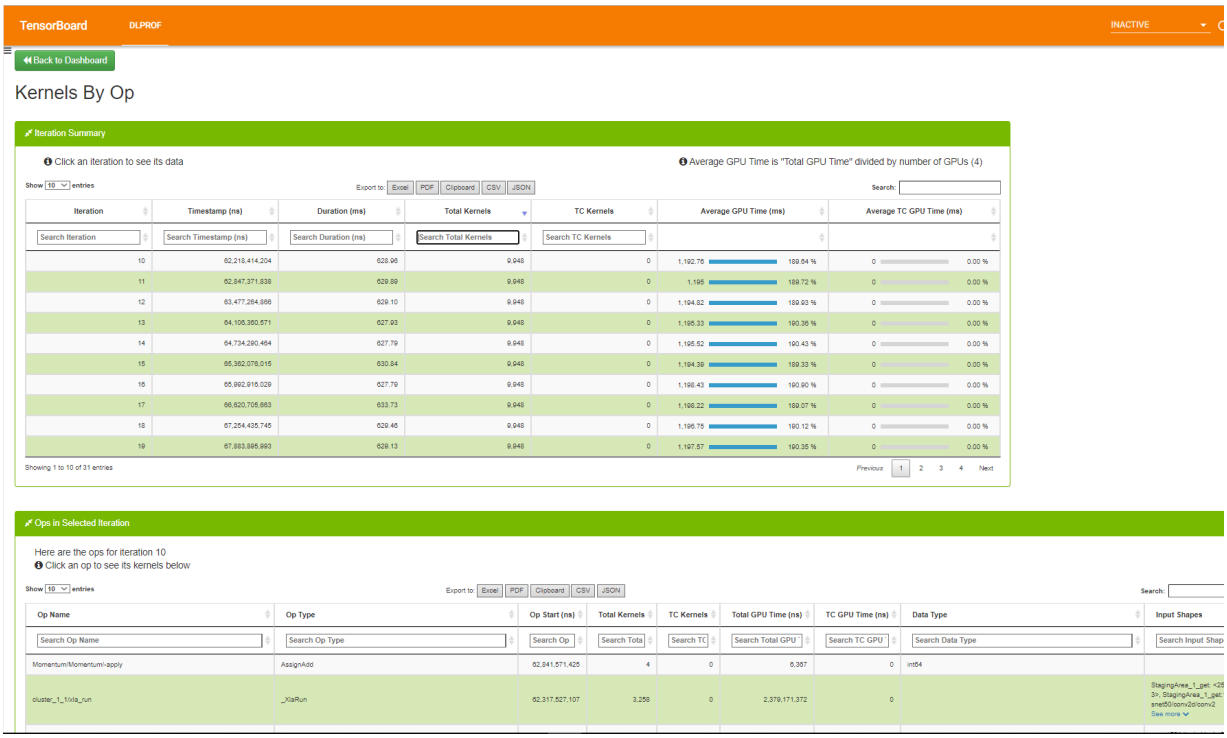
Column	Description
Op Name	The name of the op that launched the kernel.
Kernel Name	The name of the kernel.

Column	Description
Device ID	the device ID of the kernel.
Kernel Timestamp (ns)	The timestamp for when the CUDA API call was made for this kernel in the CPU thread. Useful to see the order in which the kernels were called.
GPU Time (ns)	The time spent executing the kernel on the GPU.
Users TC	A true/false field indicating whether or not the kernel uses Tensor Cores. To filter, type '1' for true, and '0' for false.
Grid	The grid size of the kernel.
Block	The block size of the kernel.

4.5. Kernels by Op

The Kernels by Op view is a variation of the Kernels by Iterations view. It has the capability to filter the list of kernels by iterations and op.

See this description for all the features available in all DataTables.



4.5.1. Iteration Summary Data Table

Selecting an iteration in the Iteration Summary table will populate the Ops in Selected Iteration table with all the profile data for the ops from the selected iteration. Selecting an op in the Ops in Selected Iteration table will populate the Kernels in Selected Op table with the list of kernels and timing data executed for the selected Op and Iteration.

See this description for all the features available in all DataTables.

Column	Description
Iteration	The iteration interval numbers
Timestamp	The exact time when this iteration started.
Duration	The length of time it took for this iteration to execute. Units on this column are dynamic.
Total Kernels	The number of GPU kernels called during this iteration.
TC Kernels	The number of GPU Tensor Core kernels called during the iteration.
Average GPU Time (µs)	Average execution time for all kernels across all GPUs during the iteration. The contents of this cell are broken into multiple pieces: 1. The value is the average execution time of this iteration across all GPUs. 2. The progress meter visually indicates how much of this iteration executed on the GPU. 3. The percentage is the average GPU time of all kernels across all GPUs over the total iteration time. 4. Hovering over this cell displays a tooltip text with helpful information. Units on this column are dynamic.
Average TC GPU Time (µs)	Average execution time for all Tensor Core kernels across all GPUs during the iteration. The contents of this cell are broken into multiple pieces: 1. The value is the average execution time of Tensor Core kernels during this iteration across all GPUs. 2. The progress meter visually indicates how much of this iteration executed Tensor Core kernels.

Column	Description
	3. The percentage is the average GPU time of kernels using Tensor Core across all GPUs over the total iteration time. 4. Hovering over this cell displays a tooltip text with helpful information. Units on this column are dynamic.

4.5.2. Ops in Selected Iteration Table

Ops in Selected Iteration

See this description for all the features available in all DataTables.

Column	Description
Op Name	The name of the op that launched the kernel.
Direction	The fprop/bprop direction of the op. (only visible on PyTorch runs).
Op Type	The type of the op.
Op Start	The time the op was launched. Used to sort ops chronologically.
Total Kernels	The number of GPU kernels called during this iteration.
TC Kernels	The number of GPU Tensor Core kernels called during the iteration.
Total GPU Time (ns)	Cumulative execution time for all kernels on the GPU during the op.
TC GPU Time (ns)	Cumulative execution time for all Tensor Core kernels on the GPU during the op.
Data Type	The data type of this op (eg, float16, int64, int32).
Input Shapes	The input shapes of the op. If the content of this cell is more than 100 characters, a 'See More' hyperlink appears. When clicked, the full contents of the cell appears. When the cell is expanded, the hyperlink text is changed to 'See Less'. When clicked, the cell collapses back to the first 100 characters.

Column	Description
Stack Trace	The stack trace of the op. (only visible on PyTorch runs) If the contents of this cell is more than 100 characters, a 'See More' hyperlink appears. When clicked, the full contents of the cell appears. When the cell is expanded, the hyperlink text is changed to 'See Less'. When clicked, the cell collapses back to the first 100 characters.

4.5.3. Kernels Selected Iteration / Op Combination Table

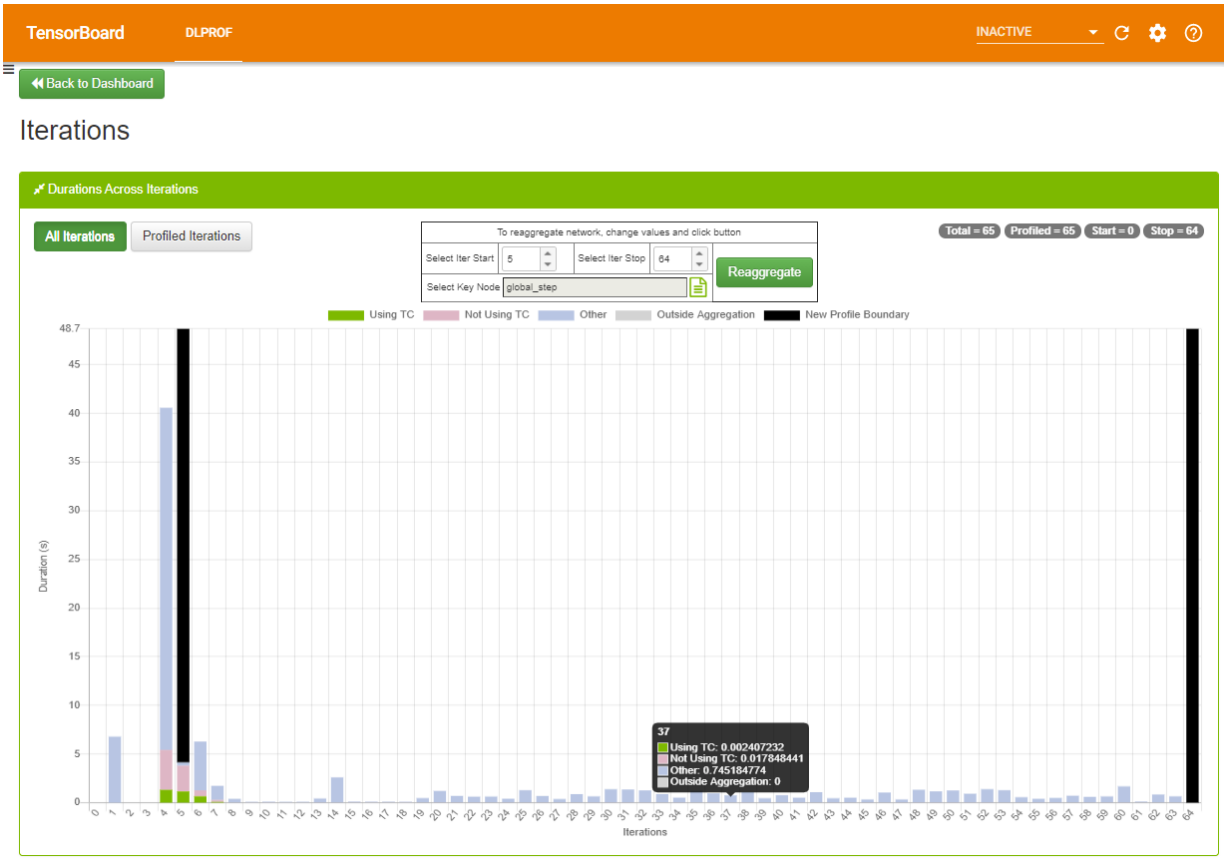
Kernels in Selected Iteration / Op combination

See this description for all the features available in all DataTables.

Column	Description
Kernel Name	The name of the kernel.
Device ID	The device ID of the kernel.
Kernel Timestamp (ns)	The timestamp for when the CUDA API call was made for this kernel in the CPU thread. Useful to see the order in which the kernels were called.
GPU Time (ns)	The time spent executing the kernel on the GPU.
Uses TC	A true/false field indicating whether or not the kernel uses Tensor Cores. To filter, type '1' for true, and '0' for false.
Grid	The grid size of the kernel.
Block	The block size of the kernel.

4.6. Iterations view

This view displays iterations visually. Users can quickly see how many iterations are in the model, the iterations that were aggregated/profiled, and the accumulated durations of tensor core kernels in each iteration. The colors on this panel match the colors on all the other dashboard panels.



Workflow

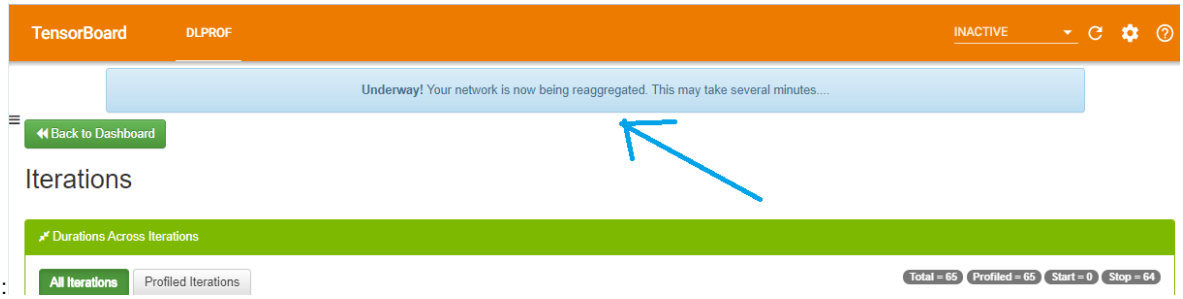
Sometimes the original aggregation parameters on the DLProf command-line specified an iteration start value, an iteration stop value, or even key node that yielded a suboptimal representation of the network's profile (such as including warm-up nodes). This feature allows the user to change those values and re-aggregate.

Here is an example:

1. Notice in the Iterations View screenshot above:
   a). Iterations two and three contain no work, and
   b). Iteration four contains way too much work. The aggregation values throughout the viewer contain those ops and kernels and potentially skews the results.

   Some platforms do not have a default key node for a neural network. Click on the 'Select Key Node' picker, to select a key node.
2. Once any of these three fields change, click the 'Reaggregate' button. After confirmation, a message is sent to the back-end DLProf server to reaggregate the profile with these values. This reaggregation process

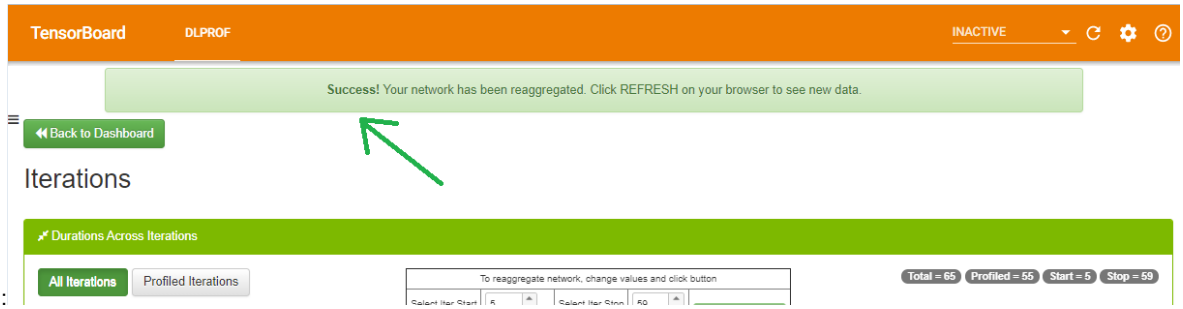
could potentially take minutes, so the following panel appears above all panels:



The viewer is fully functional while the re-aggregation is taking place. The "Underway!" panel will remain in view above all other views and panels until reaggregation is complete.

Note: Do not click the browser's BACK or REFRESH button. If either are accidentally, the "Underway!" panel will no longer appear. The re-aggregation will continue.

3. When the reaggregation is complete, a "Success!" panel will appear like this:



4. At this point, clicking the REFRESH button will load all the newly reaggregated data into the viewer.

Fields

UI Element	Definition
X-Axis	The iterations of the model.
Y-axis	Duration of iterations. The unit of this axis is dynamic (e.g., ms, ns, s, etc).
Badges	In the top right, four badges display iteration specifics of the profiled network. ▶ Total ▶ Profiled ▶ Start ▶ ttop The values inside the start and stop badges come directly from DLProf, see the `--iter_start` and `--iter_stop` command line options.
Legend	▶ Green: Accumulated duration of all kernels using tensor cores. ▶ Pink: Accumulated duration of all kernels not using tensor cores. ▶ Blue: Accumulated duration of all other kernels. ▶ Gray: Duration of iteration outside the profile.

	▶ Black: User-positioned aggregation boundary. See 'Select Iter Start' and 'Select Iter Stop' below.

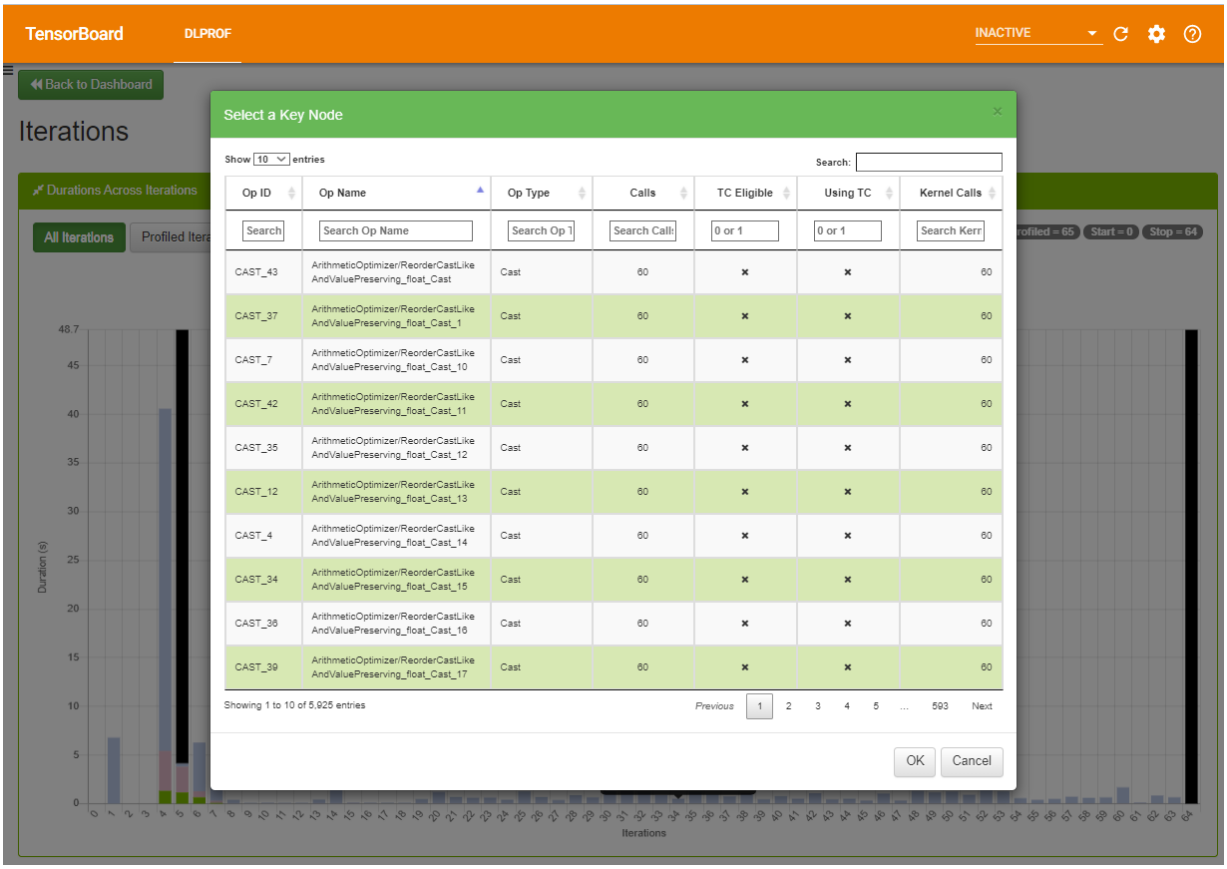
UI Controls

Control	Definition
Toggle Buttons	▶ All Iterations: Show all the iterations in the entire network. ▶ Profiled Iterations: Zooms in to only show the iterations that were profiled.
Hover	When mouse hovers over an iteration, a popup window appears displaying the values of all the constituents of the iteration.
Select Iter Start	Specification of a new "iteration start" value. Used for re-aggregating the data inside the viewer. The value can be changed by typing into the field, by clicking the up/down arrow spinners, or by hovering the cursor over the field and spinning the mouse wheel. When this value changes, the left black bar will move accordingly.
Select Iter Stop	Specification of a new "iteration stop" value. Used for re-aggregating the data inside the viewer. The value can be changed by typing into the field, by clicking the up/down arrow spinners, or by hovering the cursor over the field and spinning the mouse wheel. When this value changes, the right black bar will move accordingly.
Select Key Node	1. The existing key node value is displayed in a read-only field. 2. Sometimes key nodes can be very long. If a long key node is selected, the first few characters of the key node will appear in the field. The entire length of the key node field will appear when the cursor hovers over the field. 3. Key node picker button. Click this button to view and pick a new key node. See dialog below.
Reaggregate	Instructs the DLProf back-end server to reaggregate the network with the specified parameters.

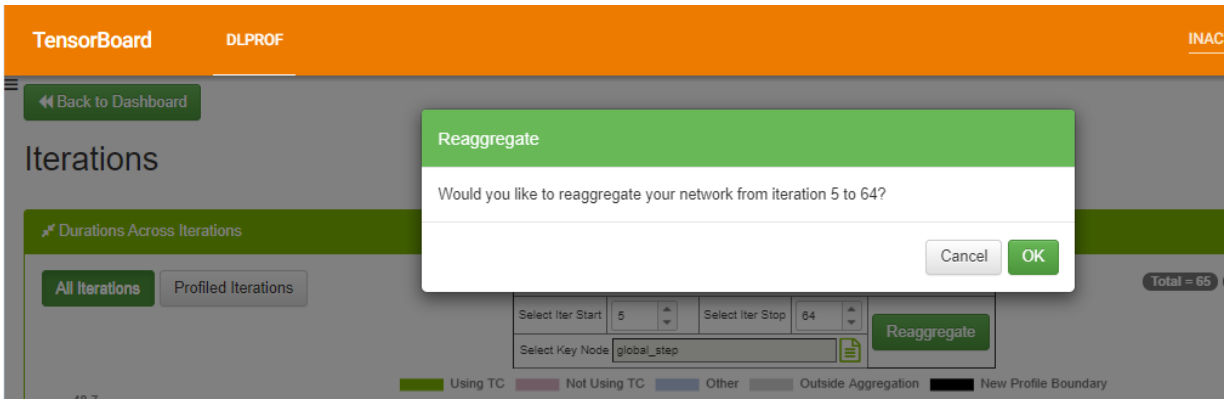
Select a Key Node Dialog

This is a full-featured panel that allows for filtering, sorting, and pagination to find a Key Node. See Ops and Kernels for more details.

Note: There are fewer columns here than in the view, but the usability is the same.



Confirmation Dialog

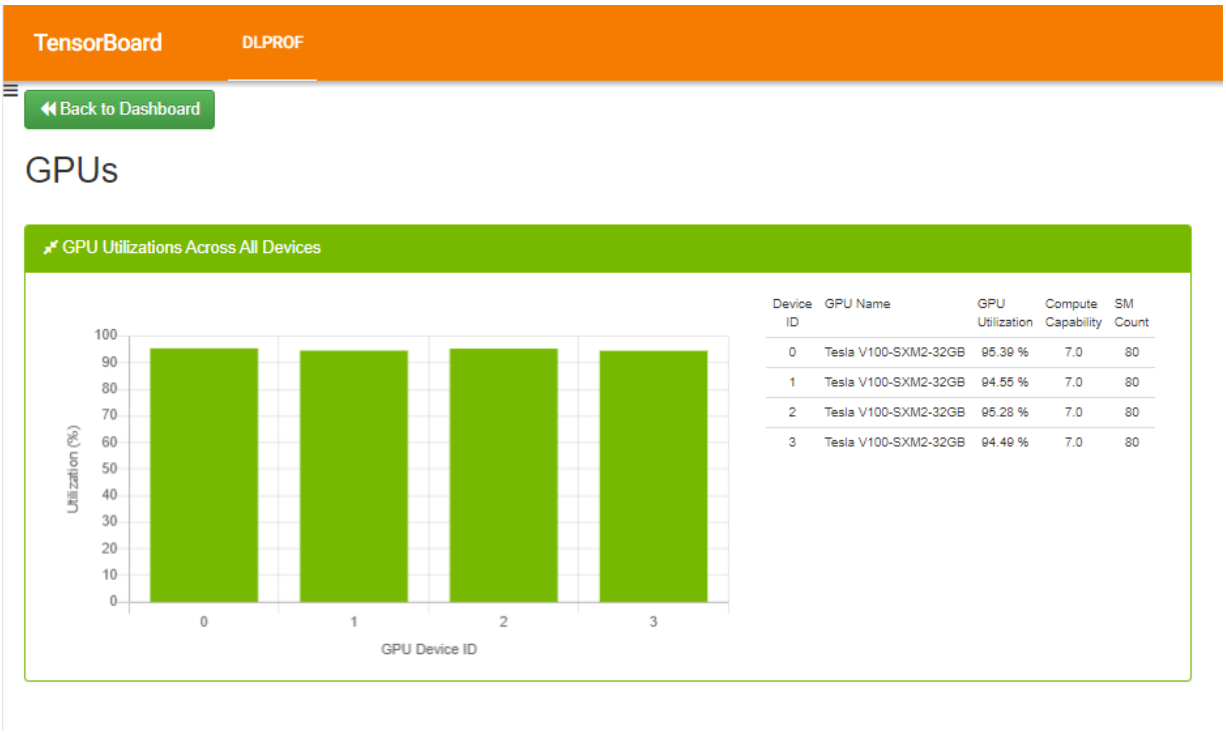


4.7. GPUs View

This view shows the utilization of all GPUs during the profile run. It is broken down into two different but related elements:

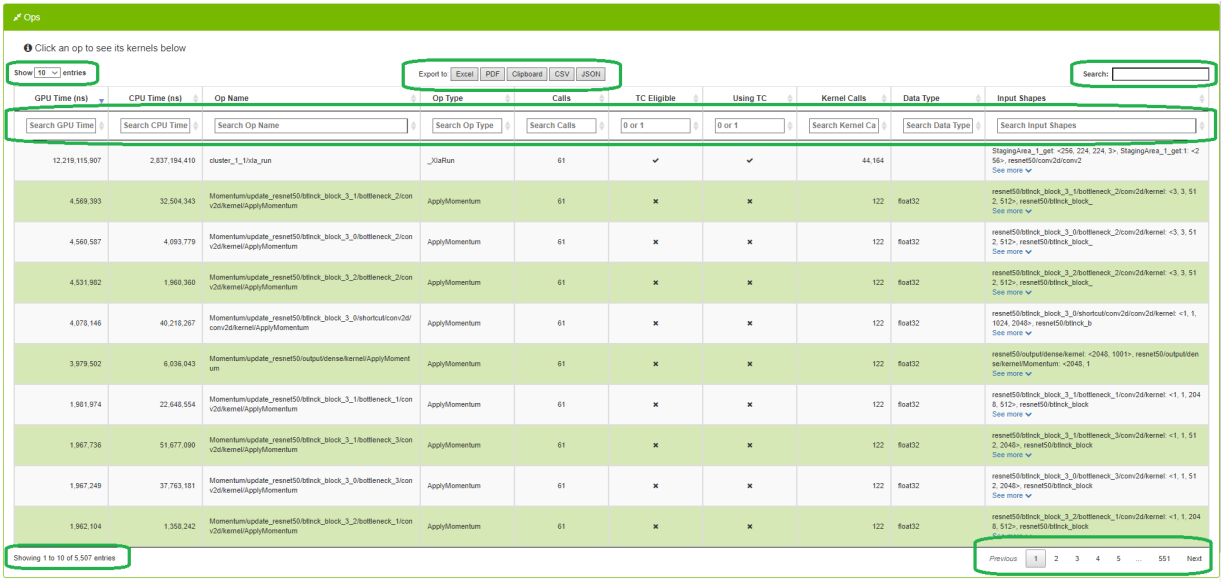
- **Bar Chart** - Quick visualization where you can see the GPU utilization for every GPU used during the profile. This view appears only when more than one GPU is used in a profile.

- **Table** - This table shows a little more detail about each GPU device, including its name, Compute Capability, and SM count.



Chapter 5. Using Data Tables

DataTables are used in many views in the DLProf Plugin. The features in DataTables enable users to quickly find information. Below is a screenshot to see the location of the features, followed by a table describing the functionality of each feature.



Data Table Feature	Definition
Showing label	The label under the table (bottom left) will show a real-time count of rows in the table.
Search text box	Filter results by text search. Typing in this field will display only those rows that contain the text in the box. Adding or removing text in the Search box will update the “Showing…” label.
Column search	Typing into the text box under most column headers will display only those rows with the entered text. This is powerful since users can enter search criteria for multiple columns to zero-in on interesting rows. Columns with ‘x’ and ‘check mark’ are boolean fields. Users can enter "1" to show those rows with a ‘check mark’, and ‘0’ to show only those rows with ‘x’.
Sort toggle button	Clicking on a column header will sort the table. When clicked, all rows are sorted either ascending or descending. The initial sort on most numeral columns are descending.
Show Entries drop list	This drop list allows the user to display 10, 25, 50, or 100 rows. Changing the setting will update the “Showing…” label.

Pagination buttons	Previous, next, and page# navigation. Allows users to quickly page through large data sets. Uses 'Show Entries' setting and updates the "Showing..." label.
Export to buttons	Allows users to easily export the data in the datatable to well known formats. An optional profile name is displayed in the header of all exports when the `--profile_name` command-line argument is used in DLProf. Warning: A slight delay occurs when any of these buttons are clicked on large networks. Enable pop-ups in your browser to export to PDF.

Chapter 6. Troubleshooting FAQ

Nothing appears in FireFox browser in TensorFlow 1.x container

Google's TensorBoard v1.15 is not compatible with Firefox version 82.0.3 (64-bit) or greater. Consequently, the DLProf plugin shows a blank or empty screen.

- **Corrective Action** Either downgrade to FireFox version 81.0.2 (64-bit) (and turn off the auto upgrade feature), or use the Chrome, Opera, or Chromium browser instead.

I see TensorBoard but not the DLPROF plugin

Multiple tensorboards are likely installed inside the container. The idea here is to remove all other TensorBoard installations and keep NVIDIA TensorBoard.

TensorFlow 2.x

pip list | fgrep -i tensorboard

```
jupyter-tensorboard             0.2.0 (<== pip uninstall this framework)
nvidia-tensorboard-plugin-dlprof          0.11 (<== keep this plugin)
tensorboard                              2.4.1 (<== keep this framework)
```

All other frameworks

pip list | fgrep -i tensorboard

```
jupyter-tensorboard              0.2.0  (<== uninstall this framework)
nvidia-tensorboard                1.15.0+nv21.1            (<== keep this framework)
nvidia-tensorboard-plugin-dlprof            0.11         (<== keep this plugin)
tensorboard                              1.15.0       (<== uninstall this framework)
```

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND 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. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com

Trademarks

NVIDIA, the NVIDIA logo, and cuBLAS, CUDA, cuDNN, DALI, DIGITS, DGX, DGX-1, DGX-2, DGX Station, DLProf, Jetson, Kepler, Maxwell, NCCL, Nsight Compute, Nsight Systems, NvCaffe, NVIDIA Ampere GPU Architecture, PerfWorks, Pascal, SDK Manager, Tegra, TensorRT, Triton Inference Server, Tesla, TF-TRT, and Volta are trademarks and/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

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com