Hauke Kirchner

# Getting started with profiling PyTorch - PointNet

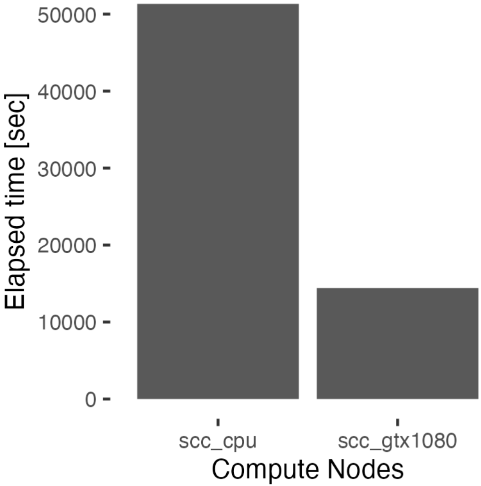
Deep Learning with GPU Cores

# Table of contents

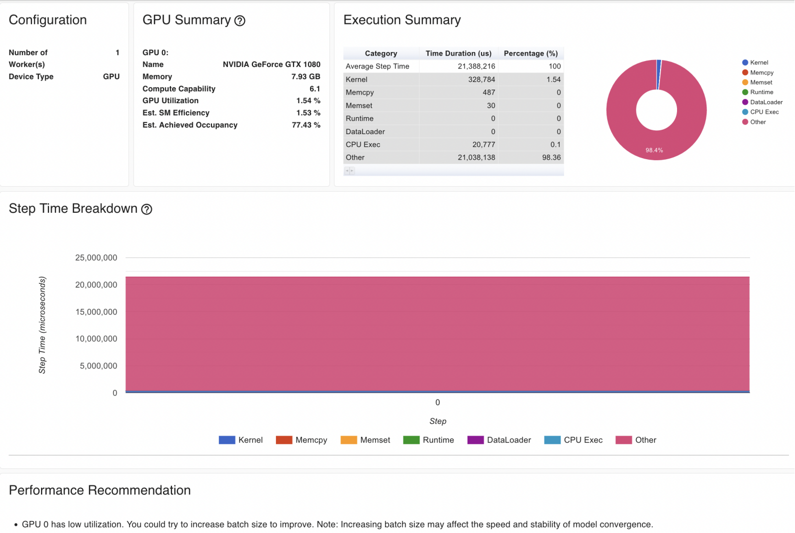
# Training setup

- number of trees for training: 8000 (740 GB)
- number of trees for testing: 2000 (191GB)
- trained for 15 epochs

# Long training times for original workflow



# Trying to use the PyTorch Profiler to find the bottleneck



## Configuration

| | |
|---|---|
| Number of Worker(s) | 1 |
| Device Type | GPU |

## GPU Summary ?

GPU 0:

| | |
|---|---|
| Name | NVIDIA GeForce GTX 1080 |
| Memory | 7.93 GB |
| Compute Capability | 6.1 |
| GPU Utilization | 1.54 % |
| Est. SM Efficiency | 1.53 % |
| Est. Achieved Occupancy | 77.43 % |

## Execution Summary

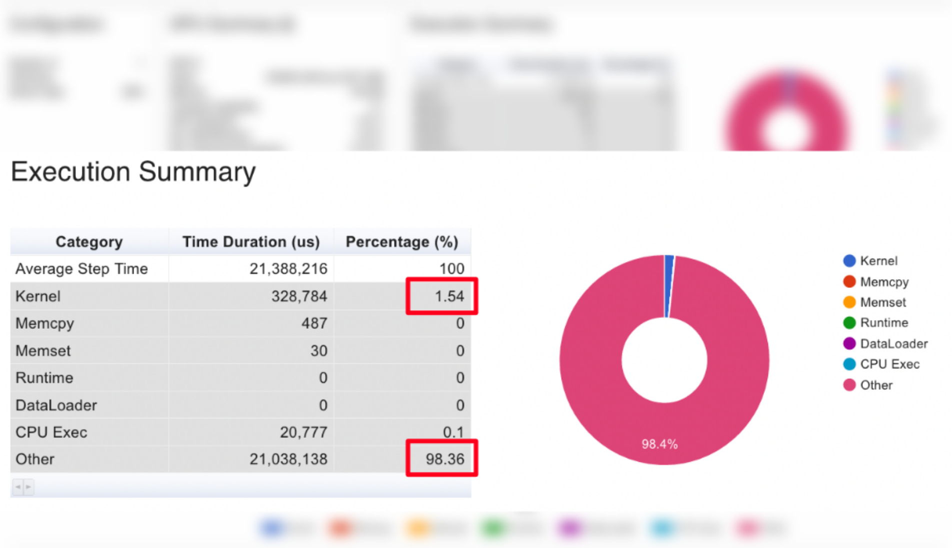
| Category | Time Duration (us) | Percentage (%) |
|---|---|---|
| Average Step Time | 21,388,216 | 100 |
| Kernel | 328,784 | 1.54 |
| Memcpy | 487 | 0 |
| Memset | 30 | 0 |
| Runtime | 0 | 0 |
| DataLoader | 0 | 0 |
| CPU Exec | 20,777 | 0.1 |
| Other | 21,038,138 | 98.36 |

## Step Time Breakdown ?

## Performance Recommendation

- GPU 0 has low utilization. You could try to increase batch size to improve. Note: Increasing batch size may affect the speed and stability of model convergence.

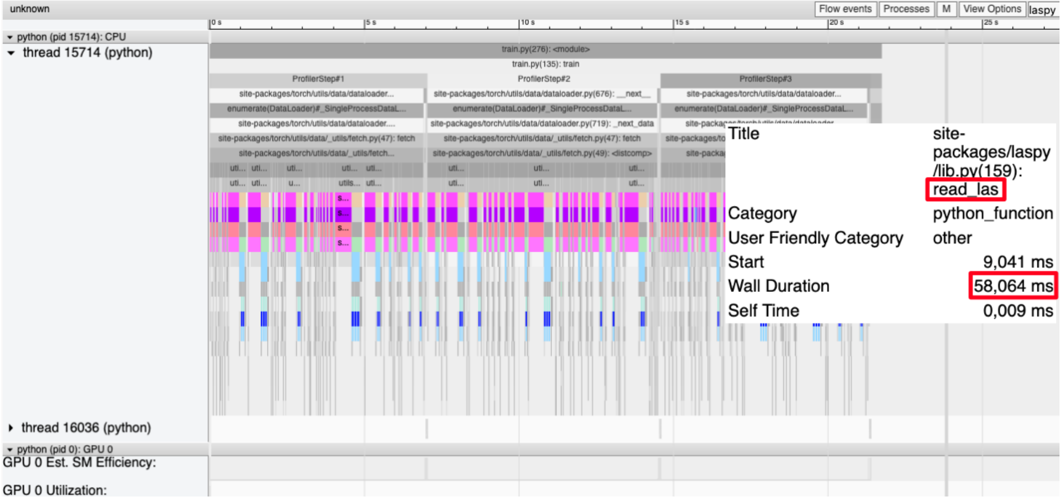
# Trying to use the PyTorch Profiler to find the bottleneck



## Execution Summary

| Category | Time Duration (us) | Percentage (%) |
|---|---|---|
| Average Step Time | 21,388,216 | 100 |
| Kernel | 328,784 | 1.54 |
| Memcpy | 487 | 0 |
| Memset | 30 | 0 |
| Runtime | 0 | 0 |
| DataLoader | 0 | 0 |
| CPU Exec | 20,777 | 0.1 |
| Other | 21,038,138 | 98.36 |

# Trace View: Laspy with raw lidar data


unknown
Flow events
Processes
M
View Options
laspy
0 s
5 s
10 s
15 s
20 s
25 s
python (pid 15714): CPU
thread 15714 (python)
train.py(276): <module>
train.py(135): train
ProfilerStep#1
ProfilerStep#2
ProfilerStep#3
site-packages/torch/utils/data/dataloader...
site-packages/torch/utils/data/dataloader.py(676): __next__
enumerate(DataLoader)#_SingleProcessDataL...
site-packages/torch/utils/data/dataloader....
site-packages/torch/utils/data/dataloader.py(719): _next_data
site-packages/torch/utils/data/_utils/fetch.py(47): fetch
site-packages/torch/utils/data/_utils/fetch...
site-packages/torch/utils/data/_utils/fetch.py(49): <listcomp>
uti...
utils...
u...
s...
thread 16036 (python)
python (pid 0): GPU 0
GPU 0 Est. SM Efficiency:
GPU 0 Utilization:

| | |
|---|---|
| Title | site-packages/laspy/lib.py(159): read_las |
| Category | python_function |
| User Friendly Category | other |
| Start | 9,041 ms |
| Wall Duration | 58,064 ms |
| Self Time | 0,009 ms |

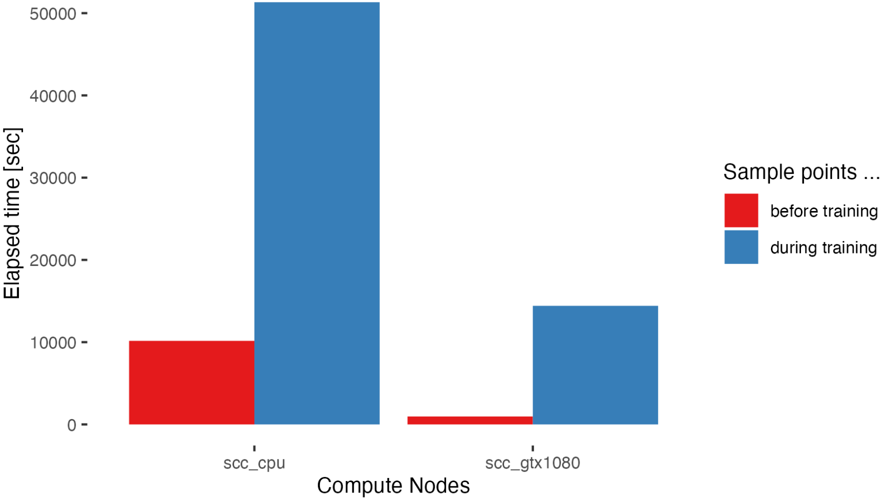
# Identification of the bottleneck

- bottleneck
  - data loading [a] and sampling process
- solution
  - sample points before training

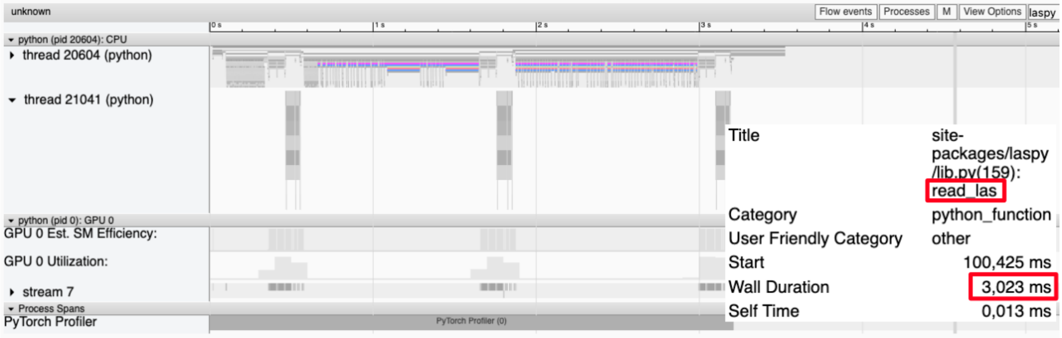
- expert knowledge about the workflow and the profiler required

[a] see "Parallel 3D Point Cloud Data analysis with Dask": https://s.gwdg.de/VRZimY

# Effect of point sampling strategies on walltime



# Trace View: Laspy with presampled lidar data



| | |
|---|---|
| Title | site-packages/laspy/lib.py(159): read_las |
| Category | python_function |
| User Friendly Category | other |
| Start | 100,425 ms |
| Wall Duration | 3,023 ms |
| Self Time | 0,013 ms |

# Training setup - improved

- number of trees for training: 8000 (~~740 GB~~ 434 MB)
- number of trees for testing: 2000 (~~191 GB~~ 109MB)
- trained for 15 epochs

---

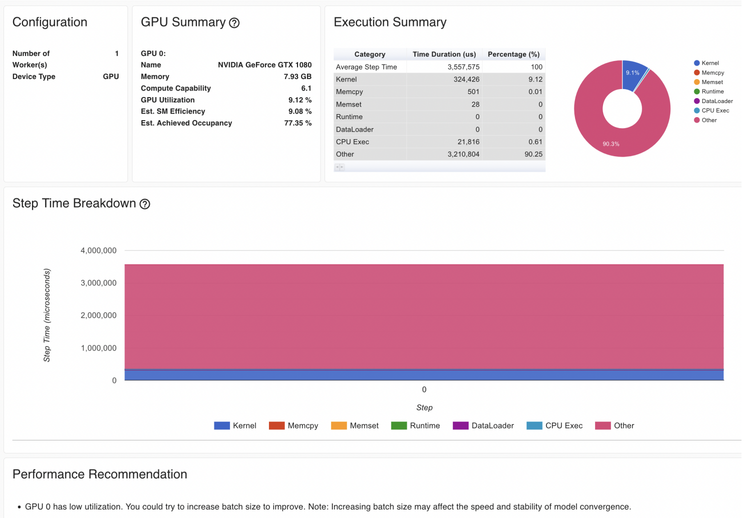
→ reduced hardware usage (energy + money savings)

→ more variations of hardware and tools can be tested

# PyTorch Profiler: Views

- **Overview**
- Operator View
- GPU Kernel View
- **Trace View**
- Memory View
- Module View

# Overview



## Configuration

| | |
|---|---|
| Number of Worker(s) | 1 |
| Device Type | GPU |

## GPU Summary ?

| | |
|---|---|
| GPU 0: | |
| Name | NVIDIA GeForce GTX 1080 |
| Memory | 7.93 GB |
| Compute Capability | 6.1 |
| GPU Utilization | 9.12 % |
| Est. SM Efficiency | 9.08 % |
| Est. Achieved Occupancy | 77.35 % |

## Execution Summary

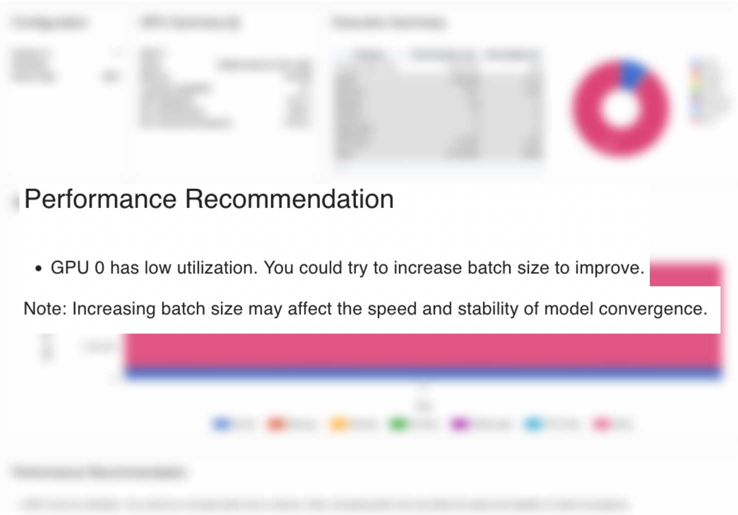
| Category | Time Duration (us) | Percentage (%) |
|---|---|---|
| Average Step Time | 3,557,575 | 100 |
| Kernel | 324,426 | 9.12 |
| Memcpy | 501 | 0.01 |
| Memset | 28 | 0 |
| Runtime | 0 | 0 |
| DataLoader | 0 | 0 |
| CPU Exec | 21,816 | 0.61 |
| Other | 3,210,804 | 90.25 |

## Step Time Breakdown ?

## Performance Recommendation

- GPU 0 has low utilization. You could try to increase batch size to improve. Note: Increasing batch size may affect the speed and stability of model convergence.

# Performance Recommendation

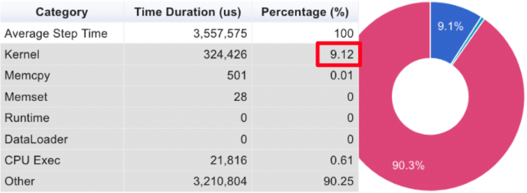


Performance Recommendation

- GPU 0 has low utilization. You could try to increase batch size to improve.

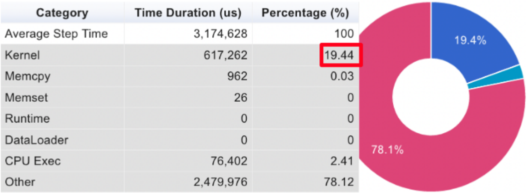
Note: Increasing batch size may affect the speed and stability of model convergence.

# Effect of increasing the batch size



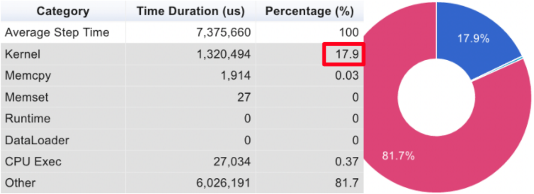
| Category | Time Duration (us) | Percentage (%) |
|---|---|---|
| Average Step Time | 3,557,575 | 100 |
| Kernel | 324,426 | 9.12 |
| Memcpy | 501 | 0.01 |
| Memset | 28 | 0 |
| Runtime | 0 | 0 |
| DataLoader | 0 | 0 |
| CPU Exec | 21,816 | 0.61 |
| Other | 3,210,804 | 90.25 |

batch size = 32



| Category | Time Duration (us) | Percentage (%) |
|---|---|---|
| Average Step Time | 3,174,628 | 100 |
| Kernel | 617,262 | 19.44 |
| Memcpy | 962 | 0.03 |
| Memset | 26 | 0 |
| Runtime | 0 | 0 |
| DataLoader | 0 | 0 |
| CPU Exec | 76,402 | 2.41 |
| Other | 2,479,976 | 78.12 |

batch size = 64



| Category | Time Duration (us) | Percentage (%) |
|---|---|---|
| Average Step Time | 7,375,660 | 100 |
| Kernel | 1,320,494 | 17.9 |
| Memcpy | 1,914 | 0.03 |
| Memset | 27 | 0 |
| Runtime | 0 | 0 |
| DataLoader | 0 | 0 |
| CPU Exec | 27,034 | 0.37 |
| Other | 6,026,191 | 81.7 |

batch size = 128

- ● Kernel
- ● Memcpy
- ● Memset
- ● Runtime
- ● DataLoader
- ● CPU Exec
- ● Other

# Outline

1 PyTorch Profiler: Optimized data loading strategy

2 PyTorch Profiler: Performance Recommendation

3 DeepSpeed - FlopsProfiler

# Summary

```
-------------------------- DeepSpeed Flops Profiler --------------------------
Profile Summary at step 5:
Notations:
data parallel size (dp_size), model parallel size(mp_size),
number of parameters (params), number of multiply-accumulate operations(MACs),
number of floating-point operations (flops), floating-point operations per second (FLOPS),
fwd latency (forward propagation latency), bwd latency (backward propagation latency),
step (weights update latency), iter latency (sum of fwd, bwd and step latency)

params per gpu:                                               3.46 M
params of model = params per GPU * mp_size:                   3.46 M
fwd MACs per GPU:                                             14.07 GMACs
fwd flops per GPU:                                            29.04 G
fwd flops of model = fwd flops per GPU * mp_size:             29.04 G
fwd latency:                                                  83.19 ms
fwd FLOPS per GPU = fwd flops per GPU / fwd latency:          349.1 GFLOPS
```

# Aggregated Profile per GPU

```
----------------------------- Aggregated Profile per GPU -----------------------------
Top 1 modules in terms of params, MACs or fwd latency at different model depths:
depth 0:
    params      - {'PointNet': '3.46 M'}
    MACs        - {'PointNet': '14.07 GMACs'}
    fwd latency - {'PointNet': '83.19 ms'}
depth 1:
    params      - {'Transform': '2.8 M'}
    MACs        - {'Transform': '14.05 GMACs'}
    fwd latency - {'Transform': '82.08 ms'}
depth 2:
    params      - {'Tnet': '2.66 M'}
    MACs        - {'Tnet': '9.34 GMACs'}
    fwd latency - {'Tnet': '58.43 ms'}
```

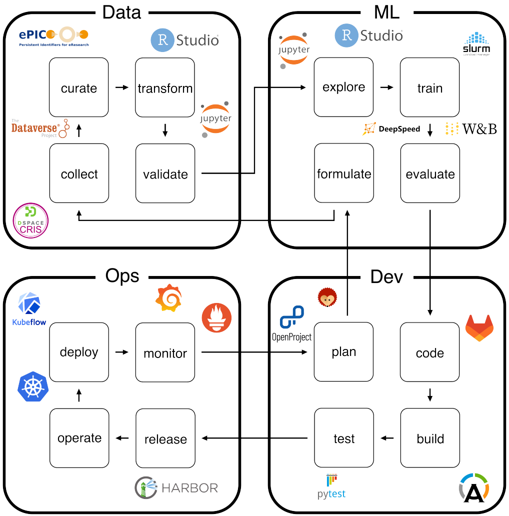
# Detailed Profile per GPU

```
------------------------------ Detailed Profile per GPU ------------------------------
Each module profile is listed after its name in the following order:
params, percentage of total params, MACs, percentage of total MACs, fwd latency, percentage of total fwd latency, fwd FLOPS

Note: 1. A module can have torch.nn.module or torch.nn.functional to compute logits (e.g. CrossEntropyLoss). They are not counted as submodules, thus not to be printed out. However they make up the difference between a parent's MACs (or latency) and the sum of its submodules'.
2. Number of floating-point operations is a theoretical estimation, thus FLOPS computed using that could be larger than the maximum system throughput.
3. The fwd latency listed in the top module's profile is directly captured at the module forward function in PyTorch, thus it's less than the fwd latency shown above which is captured in DeepSpeed.

PointNet(
  3.46 M, 100.00% Params, 14.07 GMACs, 100.00% MACs, 83.19 ms, 100.00% latency, 349.1 GFLOPS,
  (transform): Transform(
    2.8 M, 80.94% Params, 14.05 GMACs, 99.85% MACs, 82.08 ms, 98.67% latency, 353.28 GFLOPS,
    (input_transform): Tnet(
      803.08 k, 23.19% Params, 4.59 GMACs, 32.63% MACs, 27.06 ms, 32.53% latency, 350.87 GFLOPS,
      (conv1): Conv1d(256, 0.01% Params, 6.29 MMACs, 0.04% MACs, 313.28 us, 0.38% latency, 46.86 GFLOPS, 3, 64, kernel_size=(1,), stride=(1,))
      (conv2): Conv1d(8.32 k, 0.24% Params, 268.44 MMACs, 1.91% MACs, 374.32 us, 0.45% latency, 1.45 TFLOPS, 64, 128, kernel_size=(1,), stride=(1,))
```

# Next



Data

curate → transform → validate

collect → curate

ML

formulate → explore → train → evaluate

Dev

plan → code → build → test

Ops

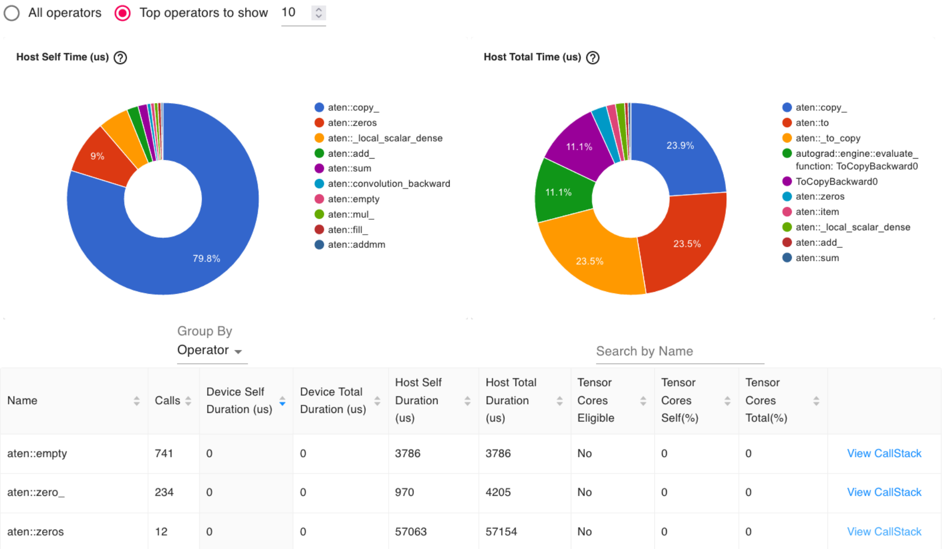
release → operate → deploy → monitor

- more workshops
  - Performance Analysis of AI and HPC Workloads
  - High Performance Data Analytics
  - Deep Learning Bootcamp: Building and Deploying AI Models
- more projects
  - KISSKI
  - NHR
- more reading
  - report and scripts for the profiling part[a]

[a] https://github.com/haukekirchner/scap

# Appendix

# Operator View



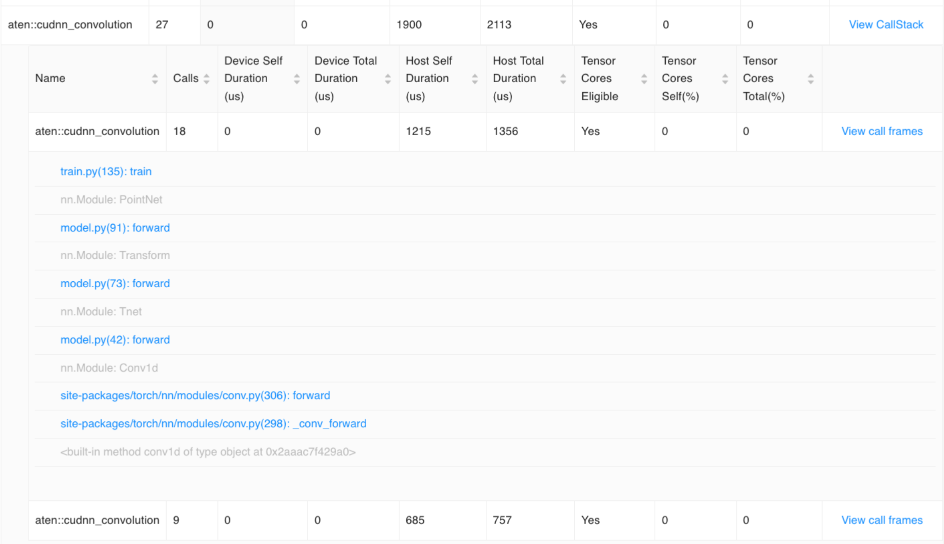
○ All operators ● Top operators to show 10

Group By
Operator

Search by Name

| Name | Calls | Device Self Duration (us) | Device Total Duration (us) | Host Self Duration (us) | Host Total Duration (us) | Tensor Cores Eligible | Tensor Cores Self(%) | Tensor Cores Total(%) | |
|---|---|---|---|---|---|---|---|---|---|
| aten::empty | 741 | 0 | 0 | 3786 | 3786 | No | 0 | 0 | View CallStack |
| aten::zero_ | 234 | 0 | 0 | 970 | 4205 | No | 0 | 0 | View CallStack |
| aten::zeros | 12 | 0 | 0 | 57063 | 57154 | No | 0 | 0 | View CallStack |

# Operator View



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| aten::cudnn_convolution | 27 | 0 | 0 | 1900 | 2113 | Yes | 0 | 0 | View CallStack |

| Name | Calls | Device Self Duration (us) | Device Total Duration (us) | Host Self Duration (us) | Host Total Duration (us) | Tensor Cores Eligible | Tensor Cores Self(%) | Tensor Cores Total(%) | |
|---|---|---|---|---|---|---|---|---|---|
| aten::cudnn_convolution | 18 | 0 | 0 | 1215 | 1356 | Yes | 0 | 0 | View call frames |

train.py(135): train

nn.Module: PointNet

model.py(91): forward

nn.Module: Transform

model.py(73): forward

nn.Module: Tnet

model.py(42): forward

nn.Module: Conv1d

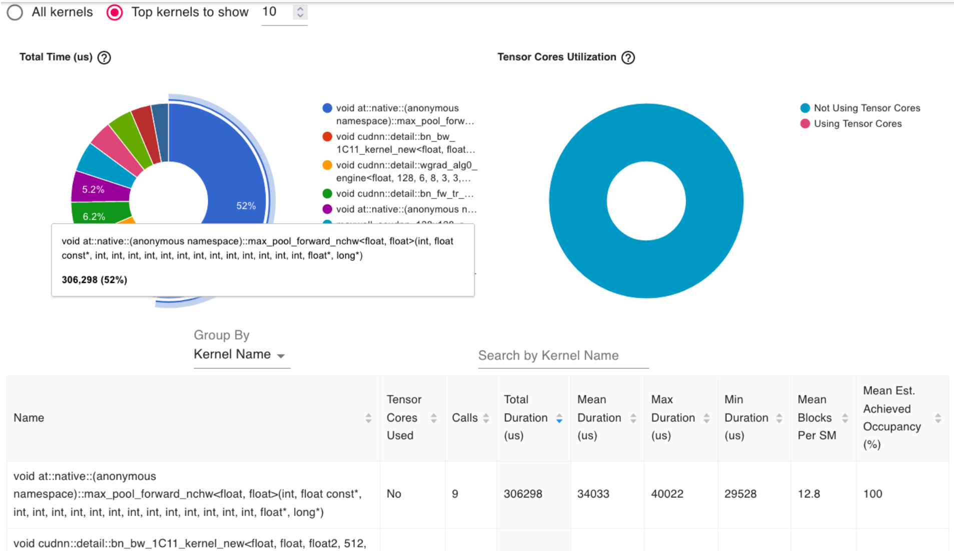
site-packages/torch/nn/modules/conv.py(306): forward

site-packages/torch/nn/modules/conv.py(298): _conv_forward

<built-in method conv1d of type object at 0x2aaac7f429a0>

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| aten::cudnn_convolution | 9 | 0 | 0 | 685 | 757 | Yes | 0 | 0 | View call frames |

# GPU Kernel View



○ All kernels ● Top kernels to show 10

Total Time (us) ?
5.2%
6.2%
52%
● void at::native::(anonymous namespace)::max_pool_forw…
● void cudnn::detail::bn_bw_1C11_kernel_new<float, float…
● void cudnn::detail::wgrad_alg0_engine<float, 128, 6, 8, 3, 3,…
● void cudnn::detail::bn_fw_tr_…
● void at::native::(anonymous n…
Tensor Cores Utilization ?
● Not Using Tensor Cores
● Using Tensor Cores

void at::native::(anonymous namespace)::max_pool_forward_nchw<float, float>(int, float const*, int, int, int, int, int, int, int, int, int, int, int, int, int, float*, long*)

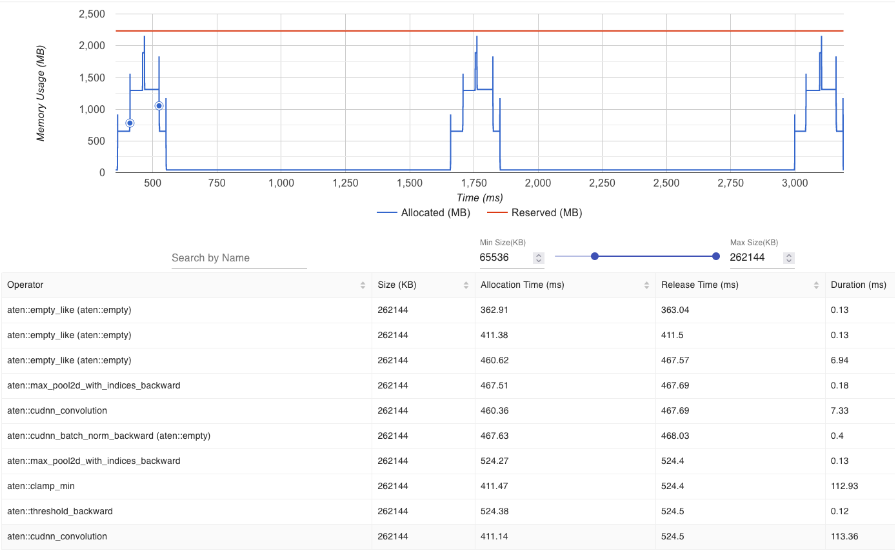
**306,298 (52%)**

Group By: Kernel Name

Search by Kernel Name

| Name | Tensor Cores Used | Calls | Total Duration (us) | Mean Duration (us) | Max Duration (us) | Min Duration (us) | Mean Blocks Per SM | Mean Est. Achieved Occupancy (%) |
|---|---|---|---|---|---|---|---|---|
| void at::native::(anonymous namespace)::max_pool_forward_nchw<float, float>(int, float const*, int, int, int, int, int, int, int, int, int, int, int, int, int, float*, long*) | No | 9 | 306298 | 34033 | 40022 | 29528 | 12.8 | 100 |
| void cudnn::detail::bn_bw_1C11_kernel_new<float, float, float2, 512, | | | | | | | | |

# Memory View



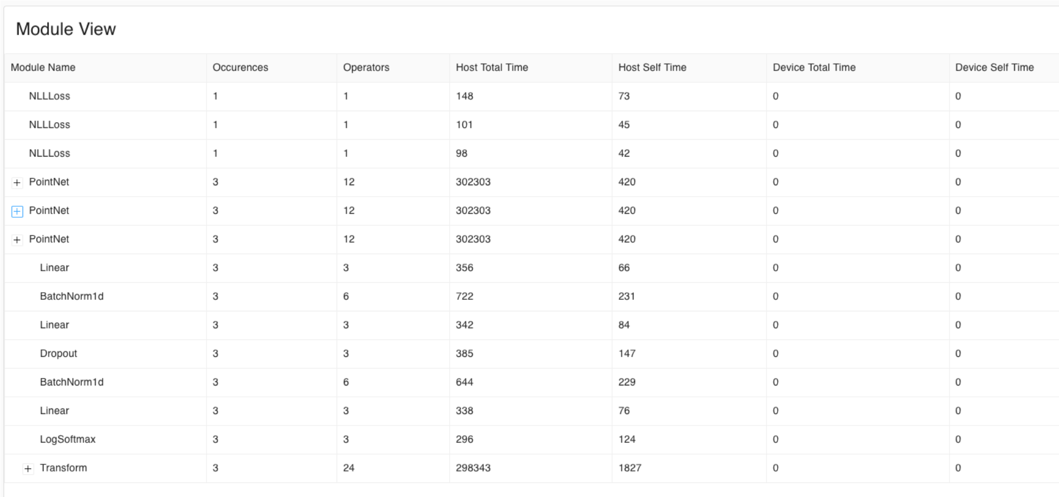
Search by Name

Min Size(KB) 65536

Max Size(KB) 262144

| Operator | Size (KB) | Allocation Time (ms) | Release Time (ms) | Duration (ms) |
|---|---|---|---|---|
| aten::empty_like (aten::empty) | 262144 | 362.91 | 363.04 | 0.13 |
| aten::empty_like (aten::empty) | 262144 | 411.38 | 411.5 | 0.13 |
| aten::empty_like (aten::empty) | 262144 | 460.62 | 467.57 | 6.94 |
| aten::max_pool2d_with_indices_backward | 262144 | 467.51 | 467.69 | 0.18 |
| aten::cudnn_convolution | 262144 | 460.36 | 467.69 | 7.33 |
| aten::cudnn_batch_norm_backward (aten::empty) | 262144 | 467.63 | 468.03 | 0.4 |
| aten::max_pool2d_with_indices_backward | 262144 | 524.27 | 524.4 | 0.13 |
| aten::clamp_min | 262144 | 411.47 | 524.4 | 112.93 |
| aten::threshold_backward | 262144 | 524.38 | 524.5 | 0.12 |
| aten::cudnn_convolution | 262144 | 411.14 | 524.5 | 113.36 |

# Module View



## Module View

| Module Name | Occurences | Operators | Host Total Time | Host Self Time | Device Total Time | Device Self Time |
|---|---|---|---|---|---|---|
| NLLLoss | 1 | 1 | 148 | 73 | 0 | 0 |
| NLLLoss | 1 | 1 | 101 | 45 | 0 | 0 |
| NLLLoss | 1 | 1 | 98 | 42 | 0 | 0 |
| + PointNet | 3 | 12 | 302303 | 420 | 0 | 0 |
| ⊞ PointNet | 3 | 12 | 302303 | 420 | 0 | 0 |
| + PointNet | 3 | 12 | 302303 | 420 | 0 | 0 |
| Linear | 3 | 3 | 356 | 66 | 0 | 0 |
| BatchNorm1d | 3 | 6 | 722 | 231 | 0 | 0 |
| Linear | 3 | 3 | 342 | 84 | 0 | 0 |
| Dropout | 3 | 3 | 385 | 147 | 0 | 0 |
| BatchNorm1d | 3 | 6 | 644 | 229 | 0 | 0 |
| Linear | 3 | 3 | 338 | 76 | 0 | 0 |
| LogSoftmax | 3 | 3 | 296 | 124 | 0 | 0 |
| + Transform | 3 | 24 | 298343 | 1827 | 0 | 0 |