Megatron Profiling Workflow

NVIDIA Profiling Toolchain

Agenda

1. NVIDIA DL Profiling Toolchain

2. Profiling Multiple GPUs
   a) with nvidia-smi (NVML)
   b) with Nsight Systems

3. Multi-Node Profiling with Nsight Systems

NVIDIA DL AND SYSTEM-LEVEL PROFILING TOOLS

Profiling workflow

Profile Application → Analyze & Inspect → Optimize

NGC GPU-optimized Docker images

PYTORCH (pytorch >=21.03)

CUDA-enabled libraries

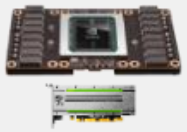
MACHINE LEARNING/ ANALYTICS: cuDF | cuML | cuGRAPH

DEEP LEARNING: cuDNN | cuBLAS | CUTLASS | NCCL | TensorRT

CuBLAS | CuFFT | OpenACC

CUDA (>=11)

Select appropriate HW with NVIDIA GPUs



Nvidia GPUs

VIRTUAL GPU

NVIDIA DGX FAMILY

NVIDIA HGX

DELL EMC, Lenovo, Hewlett Packard Enterprise

SYSTEM OEM

aws, Google Cloud Platform, Microsoft Azure

CLOUD

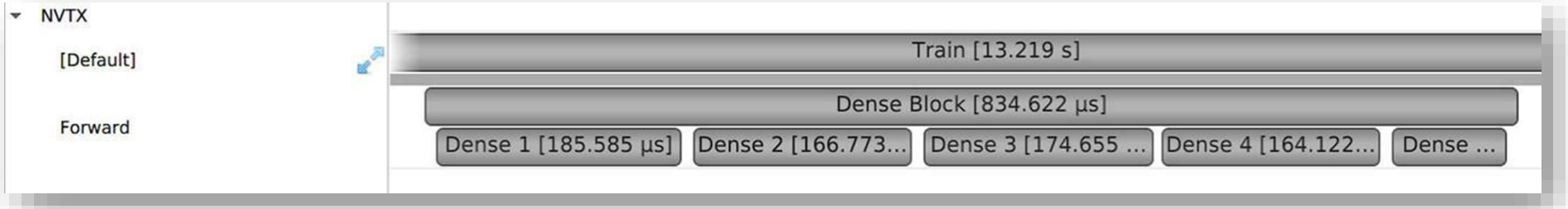
NVIDIA DL and System-Level Profiling Tools

Deep Learning Profiler (DLProf)

Nsight Systems

… captures NVTX annotations in DL frameworks and libraries

NVTX annotations in Nsight Systems timeline



NSIGHT SYSTEMS IN CONTAINERS

Enable Nsight Systems CPU sampling in the container.

docker run -it -- rm --cap-add=SYS_ADMIN -p *<port_num>:<port_num>* -p 6006:6006 -p 6007:6007 -v *path_to_local_folder*:/workspace *nvcr.io/nvidia/pytorch:21.03-py3*

Run `nsys status -e` to check sampling support.

```
root@bumblebee:/workspace# nsys status -e

Sampling Environment Check
Linux Kernel Paranoid Level = -1: OK
Linux Distribution = Ubuntu
Linux Kernel Version = 5.4.0-72-generic: OK
Linux perf_event_open syscall available: OK
Sampling trigger event available: OK
Intel(c) Last Branch Record support: Not Available
Sampling Environment: OK
```

Profiling Multiple GPUs with Nsight Systems and nvidia-smi

Megatron GPT size : small - medium

High-Level Profiling with nvidia-smi

During profiling session in real time

watch -n 1 nvidia-smi

dlprof_naive_run.sh | zcharpy@dgx0180: ~/Megat

```
 > loading shuffle-idx mapping from /home/zcharpy/Megatron-LM/dataset/SV/SV_GPT3_56kvocab_CC100Sprakbank_text_do
cument_valid_indexmap_160ns_512sl_1234s_shuffle_idx.npy
    loaded indexed file in 0.011 seconds
    total number of samples: 592450
    total number of epochs: 1
 > WARNING: could not find index map files, building the indices on rank 0 ...
 > only one epoch required, setting separate_last_epoch to False
 > elasped time to build and save doc-idx mapping (seconds): 0.081182
    using:
     number of documents:       222323
     number of epochs:          1
     sequence length:           512
     total number of samples:   14981
 > elasped time to build and save sample-idx mapping (seconds): 0.005381
 > building shuffle index with split [0, 14981) and [14981, 14981) ...
 > elasped time to build and save shuffle-idx mapping (seconds): 0.002913
 > loading doc-idx mapping from /home/zcharpy/Megatron-LM/dataset/SV/SV_GPT3_56kvocab_CC100Sprakbank_text_docume
nt_test_indexmap_80ns_512sl_1234s_doc_idx.npy
 > loading sample-idx mapping from /home/zcharpy/Megatron-LM/dataset/SV/SV_GPT3_56kvocab_CC100Sprakbank_text_doc
ument_test_indexmap_80ns_512sl_1234s_sample_idx.npy
 > loading shuffle-idx mapping from /home/zcharpy/Megatron-LM/dataset/SV/SV_GPT3_56kvocab_CC100Sprakbank_text_do
cument_test_indexmap_80ns_512sl_1234s_shuffle_idx.npy
    loaded indexed file in 0.012 seconds
    total number of samples: 14982
    total number of epochs: 1
> finished creating GPT datasets ...
time (ms) | model-and-optimizer-setup: 1359.98 | train/valid/test-data-iterators-setup: 67871.82
[after dataloaders are built] datetime: 2021-08-19 07:25:51
done with setup ...
training ...
[before the start of training step] datetime: 2021-08-19 07:25:51
[Rank 0] (after 10 iterations) memory (MB) | allocated: 5500.90869140625 | max allocated: 7316.13720703125 | res
erved: 10214.0 | max reserved: 10214.0
 iteration       10/     100 | consumed samples:           80 | elapsed time per iteration (ms): 4990.8 | learni
ng rate: 1.471E-04 | global batch size:     8 | lm loss: 9.957390E+00 | loss scale: 1.0 | grad norm: 1.720 | num
ber of skipped iterations:   0 | number of nan iterations:   0 |
time (ms) | forward-compute: 3170.80 | backward-compute: 1415.31 | backward-params-all-reduce: 203.06 | backward
-embedding-all-reduce: 0.15 | optimizer: 182.47 | batch-generator: 351.04
```

zcharpy@dgx0180: ~

```
Every 1.0s: nvidia-smi                                  dgx0180: Thu Aug 19 07:26:3
Thu Aug 19 07:26:39 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.51.05    Driver Version: 450.51.05    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000000:06:00.0 Off |                    0 |
| N/A   38C    P0    66W / 300W |  11820MiB / 16160MiB |     89%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  On   | 00000000:07:00.0 Off |                    0 |
| N/A   40C    P0    73W / 300W |  11828MiB / 16160MiB |      6%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-SXM2...  On   | 00000000:0A:00.0 Off |                    0 |
| N/A   40C    P0   127W / 300W |  11828MiB / 16160MiB |     14%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-SXM2...  On   | 00000000:0B:00.0 Off |                    0 |
| N/A   38C    P0    92W / 300W |  11788MiB / 16160MiB |     16%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   4  Tesla V100-SXM2...  On   | 00000000:85:00.0 Off |                    0 |
| N/A   37C    P0    87W / 300W |  11788MiB / 16160MiB |     12%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   5  Tesla V100-SXM2...  On   | 00000000:86:00.0 Off |                    0 |
| N/A   38C    P0    63W / 300W |  11868MiB / 16160MiB |     34%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   6  Tesla V100-SXM2...  On   | 00000000:89:00.0 Off |                    0 |
| N/A   44C    P0    74W / 300W |  11868MiB / 16160MiB |     34%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   7  Tesla V100-SXM2...  On   | 00000000:8A:00.0 Off |                    0 |
```

Profiling with Nsight Systems

```
nsys profile --duration=300 --trace=cuda,nvtx \
-o $PROFILE_OUTPUT_PATH --force-overwrite=true \
```

nsys decoration

```
python -m torch.distributed.launch \
$DISTRIBUTED_ARGS pretrain_gpt.py $GPT_ARG
```

Normal Megatron training launch

Open Nsight Systems GUI and Monitor on Berzelius

x_zench@berzelius001 - ThinLinc Client

Applications Xfce Terminal Wed 15 Sep, 12:41 Zenodia Charp

Terminal - x_zench@berzelius001:/proj/guest_at_nsc/users/zcharpy

File Edit View Terminal Tabs Help

```
[I 12:27:33.384 LabApp] EOF on FD 85; stopping reading
[I 12:27:33.484 LabApp] Terminal 1 closed
Singularity> exit
exit
[x_zench@node022 zcharpy]$ exit
exit
[x_zench@berzelius001 zcharpy]$ ls
2N16A100.sh                          loopnodes.sh
8gpu_run.sh                          Megatron-LM
check_account.sh                     multinode_run.sh
dataset                              mycert.pem
Day1-runMegatron-LM_GPT_template.sh  output
download_NGCpytorch_sif.sh           param_cnt.sh
get_certificate.sh                   pytorch_21.03.sif
GPT_43B.sh                           singularityNoDistributedLaunch.sh
gpubootcamp                          test_singularity.sh
[x_zench@berzelius001 zcharpy]$ cp gpubootcamp/ai/Megatron/English/Python/source
_code/Day1-runMegatron-LM_GPT_template.sh ../assets/
[x_zench@berzelius001 zcharpy]$ ls ../assets/
dataset
Day1-runMegatron-LM_GPT_template.sh
NsightSystems-linux-public-2021.3.1.54-ee9c30a.run
pytorch_21.03.sif
[x_zench@berzelius001 zcharpy]$
```

Terminal - x_zench@berzelius001:/proj/guest_at_nsc/users/assets

File Edit View Terminal Tabs Help

```
 1000  mkdir assets
 1001  ls
 1002  cp ./zcharpy/pytorch_21.03.sif ./assets/
 1003  cp -r ./zcharpy/dataset ./assets/
 1004  cd assets/dataset/
 1005  ls
 1006  rm download_*
 1007  ls
 1008  clear
 1009  cd ..
 1010  ls
 1011  wget https://developer.download.nvidia.com/devtools/nsight-systems/Nsight
Systems-linux-public-2021.3.1.54-ee9c30a.run
 1012  ls
 1013  rm NsightSystems-linux-public-2021.3.1.54-ee9c30a.run
 1014  ls
 1015  wget https://developer.download.nvidia.com/devtools/nsight-systems/Nsight
Systems-linux-public-2021.3.1.54-ee9c30a.run
 1016  ls
 1017  bash NsightSystems-linux-public-2021.3.1.54-ee9c30a.run
 1018  nsys --version
 1019  /proj/guest_at_nsc/users/nsight-systems-2021.3.1/bin/nsys-ui
 1020  history
[x_zench@berzelius001 assets]$
```

/assets

```
[x_zench@berzelius001 ~]$ cd /proj/guest_at_nsc/users/assets/
[x_zench@berzelius001 assets]$ pwd
/proj/guest_at_nsc/users/assets
[x_zench@berzelius001 assets]$ ls
dataset  NsightSystems-linux-public-2021.3.1.54-ee9c30a.run  pytorch_21.03.sif
[x_zench@berzelius001 assets]$
```

NVIDIA

File Edit View Run Kernel Tabs Settings Help

profile_2nd_run.sh | Lab1-6_Observe_GPT_runs | profile_naive_run.sh

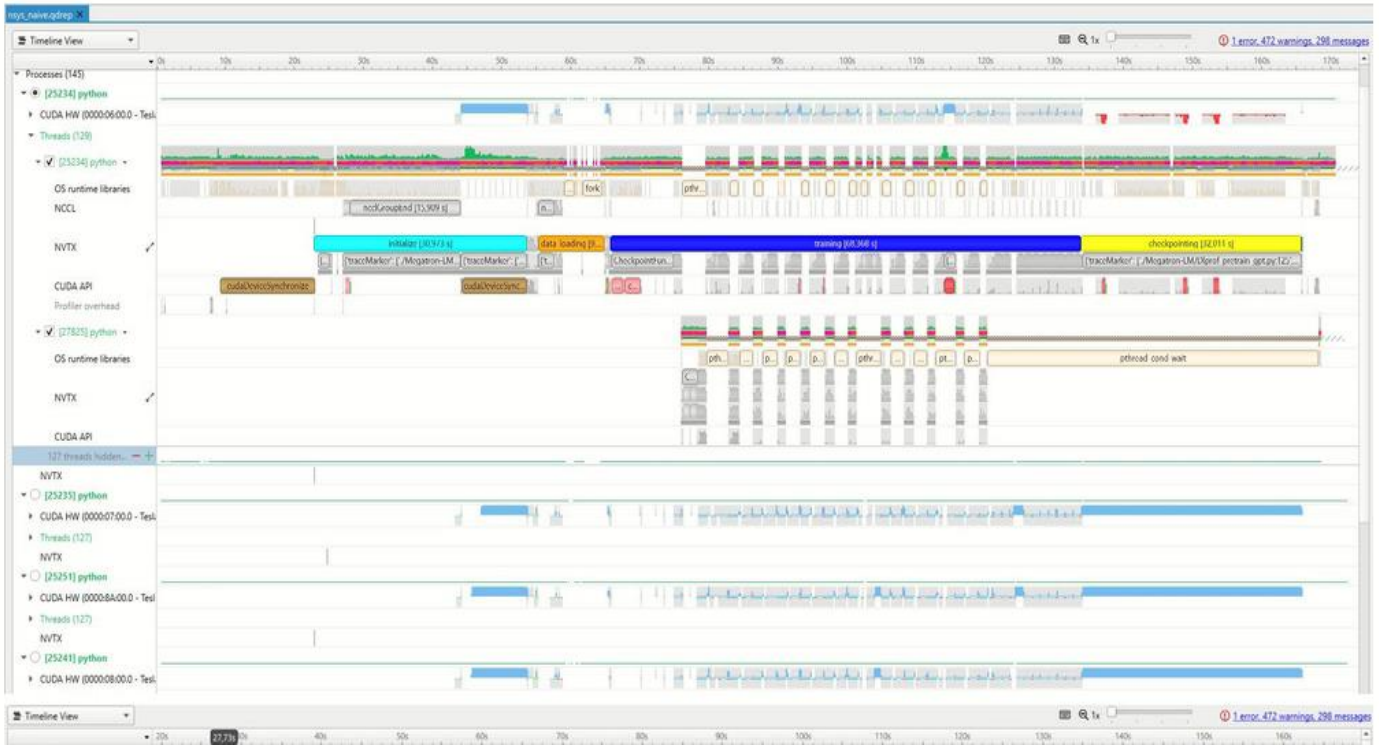
Markdown | Python 3

modify the naive run bash script directly

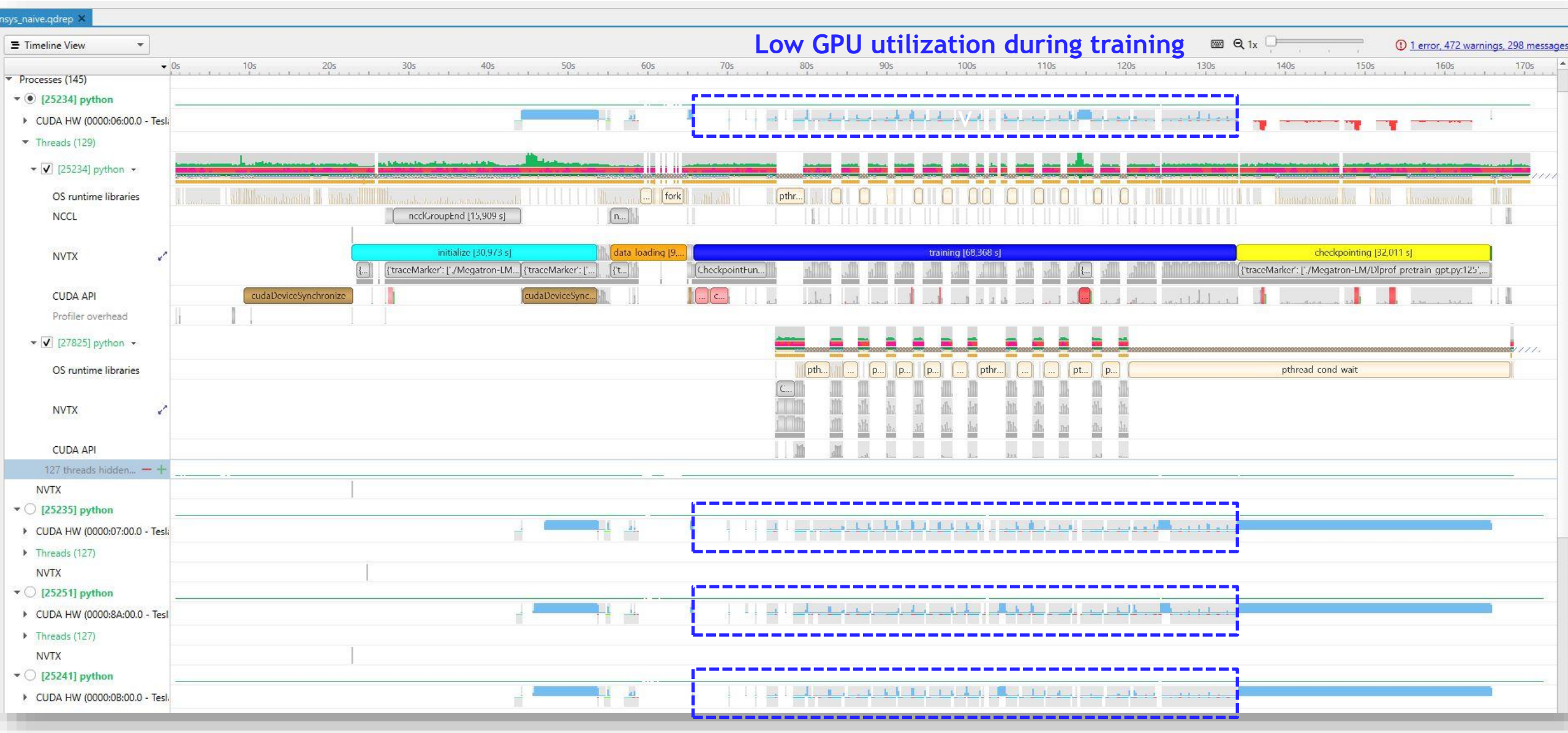
open profile_naive_run.sh

```
[ ]: !bash ./Megatron-LM/profile_naive_run.sh
```

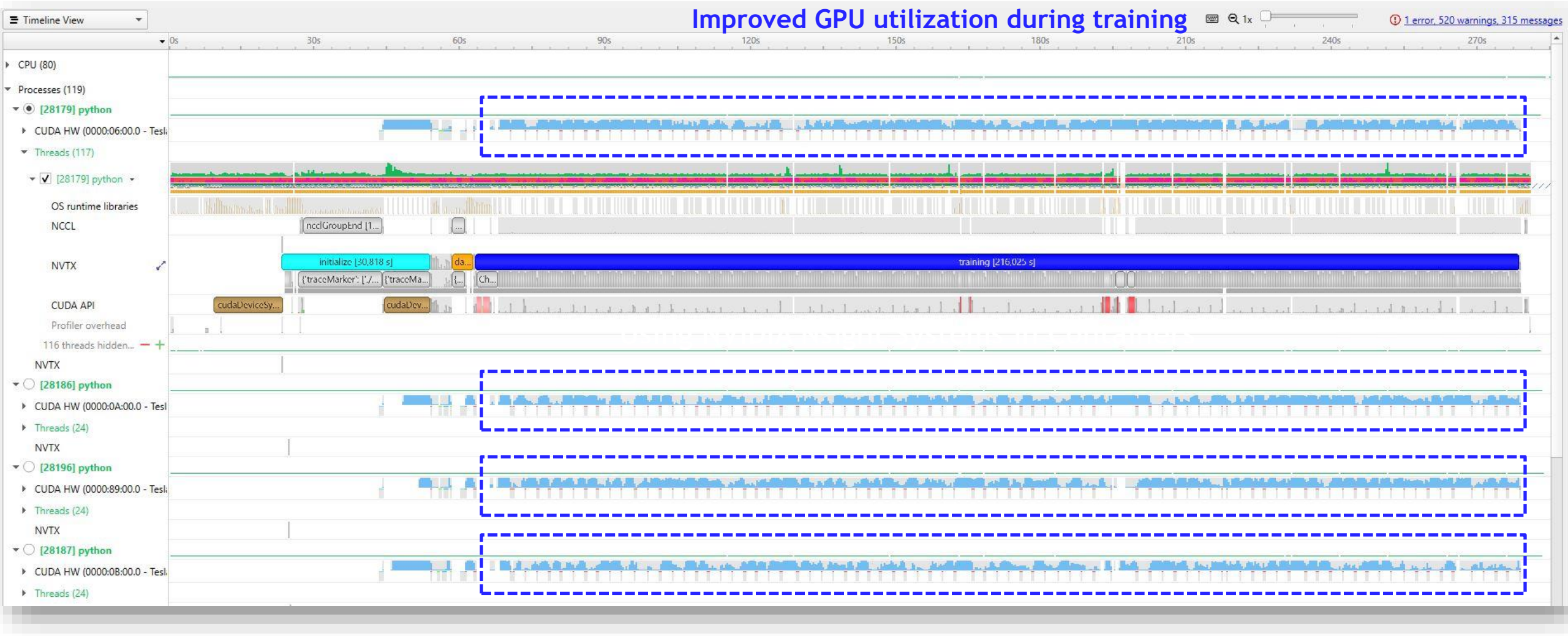
visualizing the profiles via nsight should look similar to the following



Nsight Systems Timeline Visualization - Naive Run with Multiple GPUs



Nsight Systems Timeline Visualization - Improved Run with Multiple GPUs



Multi-Node Profiling with Nsight Systems

Megatron GPT size : medium - large

Profiling with Nsight Systems

Modifications are necessary accoridng to each cluster custom setups

```
nsys profile --delay 30 --duration=1600 --kill=none \
 --trace=cudnn,cuda,nvtx --stats=true \
-o ${DIR}/profiles/name_%q{SLURM_NODEID}_%q{SLURM_LOCALID} \
--force-overwrite=true \
```

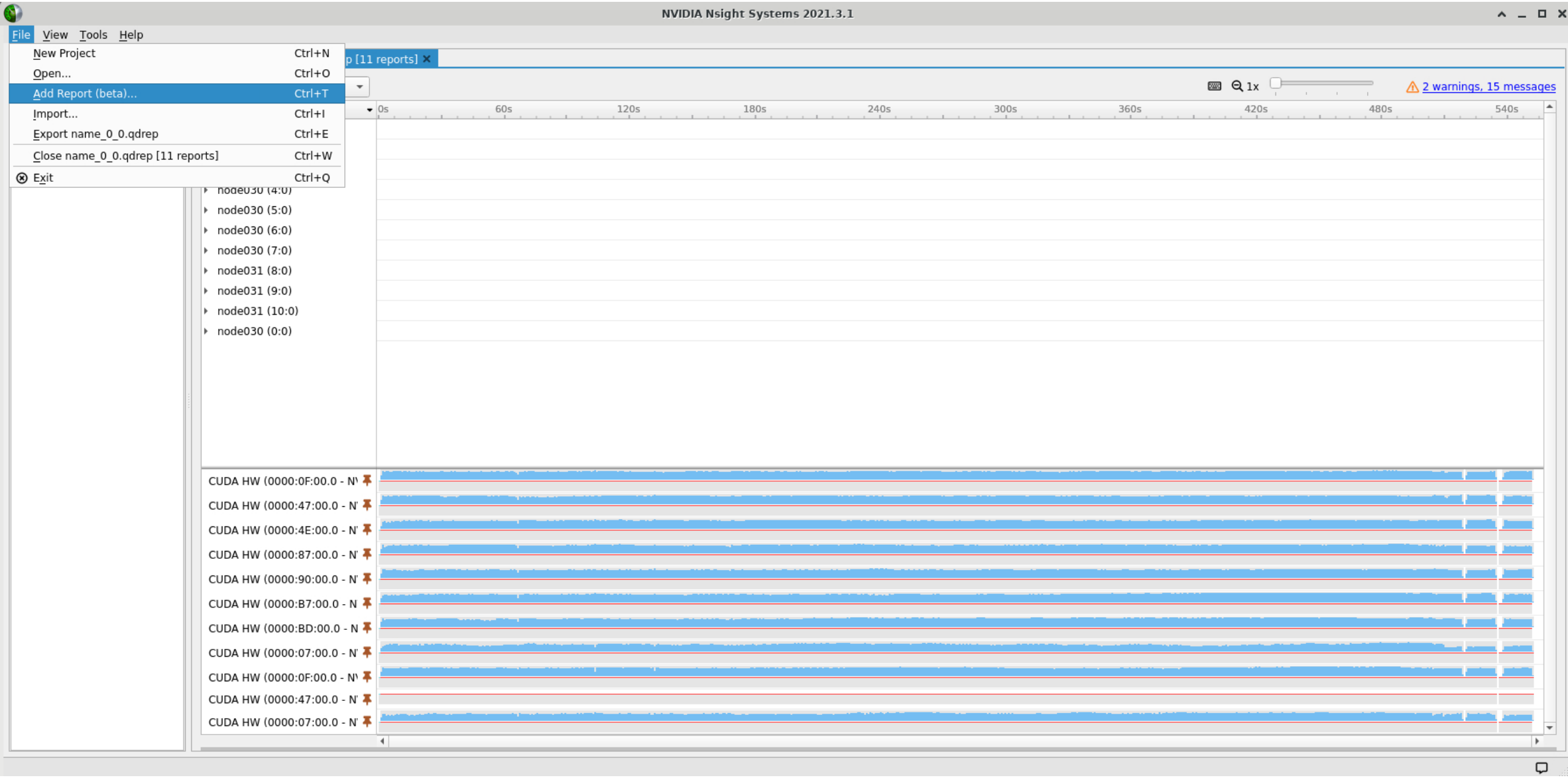
nsys decoration

```
python -u ${DIR}/Megatron-LM/pretrain_gpt.py $@ $GPT_ARGS $OUTPUT_ARGS \
--save $CHECKPOINT_PATH --load $CHECKPOINT_PATH --data-path $DATA_PATH
```

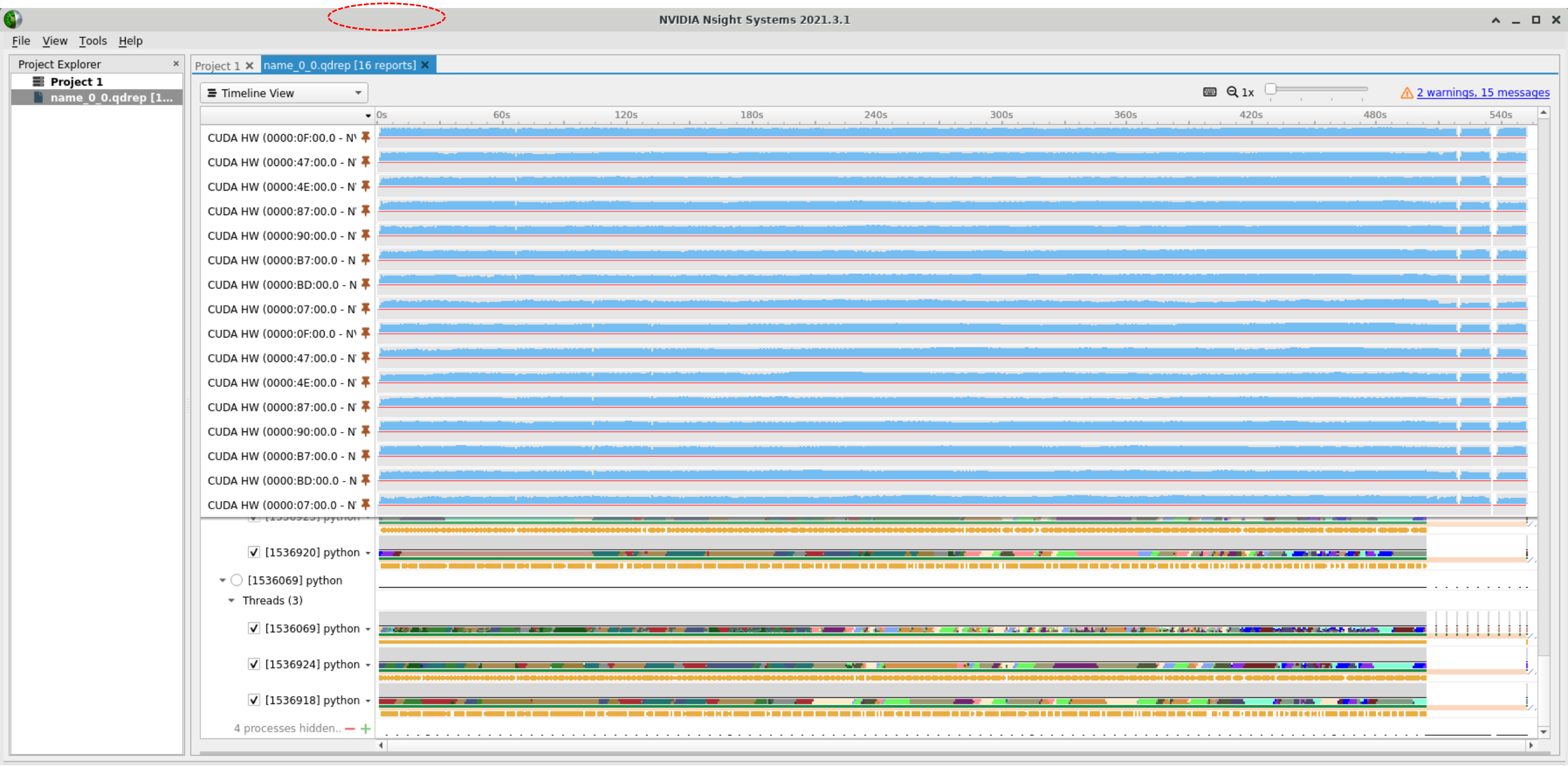
Normal Megatron training launch

Profiling Multi-node Training [Demo]

Add report files into existing timeline



16 GPUs training profiles in a single timeline



THANK YOU!

Download	https://developer.nvidia.com/nsight-systems **NOTE: website version is newer than CUDA Toolkit version**
Docs	https://docs.nvidia.com/nsight-systems/index.html
Forums	https://devtalk.nvidia.com
Email	nsight-systems@nvidia.com
Blogs	https://developer.nvidia.com/blog/nvidia-nsight-systems-containers-cloud/ https://developer.nvidia.com/blog/nsight-systems-exposes-gpu-optimization/ https://developer.nvidia.com/blog/understanding-the-visualization-of-overhead-and-latency-in-nsight-systems/

Backup:
Profiling 1 GPU - DLProf

Megatron GPT size : small

Profiling With DLProf

```
dlprof -y=60 -d=360 --mode='pytorch' --force=true \
--output_path=/home/zcharpy/profiles/DLprof/naive/ \
--profile_name='GPT360M_naive' \
--iter_start 0 --iter_stop 2 --dump_model_data=true \
--reports=summary,detail,op_type,expert_systems \
python -m torch.distributed.launch $DISTRIBUTED_ARGS \
    pretrain_gpt.py $GPT_ARG
```

DLProf decoration

Normal Megatron training run

TensorBoard DLPROF

Profiling – first trial naive run (1 GPU)

INACTIVE

GPU Utilization (%)

- Utilized
- Idle

Total Op GPU Time (ms)

- Ops Using TC
- Ops Eligible For TC
- All Other Ops

More...

Total Kernel GPU Time (ms)

- Kernels Using TC
- Memory Kernels
- All Other Kernels

More...

Tensor Core Kernel Efficiency (%)

- Using TC
- Not Using TC

Performance Summary

Wall Clock Time (min)	1.64
Tensor Core Kernel Efficiency (%)	42.5
GPU Utilization (%)	28.1
Total Iterations	1
Profiled Iterations	1
Start Iteration	0
Stop Iteration	0
Average Iteration Time (min)	1.64

Iteration Summary

More...

Top 10 GPU Ops

GPU Time (ns)	Op Name	Op Type	Calls	TC Eligible	Using TC
3,073,524,010	/module/pretrain/train/train_step/no_grad::decorate_context/FP32Optimizer::step/wrapper/step	step	100	✖	✖
2,121,781,520	/module/pretrain/save_checkpoint/save	save	1	✖	✖
2,113,616,610	/module/pretrain/train/save_checkpoint_and_time/save_checkpoint/save	save	1	✖	✖
1,613,730,568	/module/pretrain/train/train_step/forward_backward_no_pipelining/forward_step/DistributedDataParallel::_call_impl/DistributedDataParallel::forward/GPTModel::_call_impl/GPTModel::forward/TransformerLanguageModel::_call_impl/TransformerLanguageModel::forward/ParallelTransformer::_call_impl(2)/ParallelTransformer::forward/ParallelTransformer::_checkpointed_forward/checkpoint/forward/ParallelTransformer::custom_forward/ParallelTransformerLayer::_call_impl/ParallelTransformerLayer::forward/ParallelMLP::_call_impl(2)/ParallelMLP::forward/ColumnParallelLinear::_call_impl/ColumnParallelLinear::forward/linear	linear	2400	✔	✖
1,611,664,207	/_VocabParallelCrossEntropyBackward::apply/backward/ParallelTransformer::custom_forward/ParallelTransformerLayer::_call_impl/ParallelTransformerLayer::forward/ParallelMLP::_call_impl(2)/ParallelMLP::forward/ColumnParallelLinear::_call_impl/ColumnParallelLinear::forward/linear	linear	2400	✔	✖
941,545,051	/module/pretrain/train/train_step/DistributedDataParallel::allreduce_gradients/copy_	copy_	29200	✖	✖
886,010,843	/_VocabParallelCrossEntropyBackward::apply/backward/ParallelTransformer::custom_forward/ParallelTransformerLayer::_call_impl/ParallelTransformerLayer::forward/ParallelMLP::_call_impl(2)/ParallelMLP::forward/RowParallelLinear::_call_impl(2)/RowParallelLinear::forward/linear	linear	1529	✔	✔
882,259,289	/module/pretrain/train/train_step/no_grad::decorate_context/FP32Optimizer::step/FP32Optimizer::clip_grad_norm/clip_grad_norm_fp32/MultiTensorApply::__call__(2)/multi_tensor_scale	multi_tensor_scale	100	✖	✖
871,612,799	/module/pretrain/train/train_step/DistributedDataParallel::allreduce_gradients/_flatten_dense_tensors/cat	cat	100	✖	✖
856,341,817	/module/pretrain/train/train_step/DistributedDataParallel::allreduce_gradients/__itrediv__	__itrediv__	100	✖	✖

System Config

Profile Name	gpt3_750m_naive
GPU Count	1
GPU Name(s)	RTX A6000
CPU Model	AMD Ryzen Threadripper 3970X 32-Core Processor
GPU Driver Version	460.73.01
Framework	PyTorch 1.9.0a0+df837d0
CUDA Version	11.2

Recommendations

Problem	Recommendation
61 ops were eligible to use tensor cores but none are using FP16	Try enabling AMP (Automatic Mixed Precision). For more information: https://developer.nvidia.com/automatic-mixed-precision
Unable to split profile into training iterations: key node not found	Specify key node by setting the --key_node argument
Slow debug APIs were enabled. When not debugging, these APIs can slow down execution of your model	Do not use the record_function decorator or context manager unless debugging.
GPU Memory is underutilized: Only 38% of GPU Memory is used	Try increasing batch size by 2x to increase data throughput

Guidance

Understanding GPU utilization and timing details of the operations is the first step in profiling your model.

- To learn more about Tensor cores and Mixed Precision training, visit this site:https://developer.nvidia.com/tensor_cores
- You will find resources on how to train networks with mixed precision and make full use of Tensor cores for Tensorflow models here: https://docs.nvidia.com/deeplearning/sdk/mixed-precision-training/index.html#training_tensorflow
- Note that if there are multiple kernels being observed on single op node, these are likely performing data transposes to prepare the data for efficient use by tensorcores. Such transposes themselves would not use tensor cores.

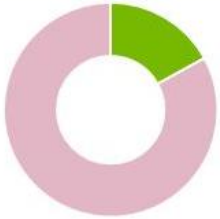
TensorBoard DLPROF

Profiling – 2nd trial applying optimization (recommanded from Dlprof)

INACTIVE

GPU Utilization (%)

- Utilized
- Idle



Total Op GPU Time (ms)

- Ops Using TC
- Ops Eligible For TC
- All Other Ops

More...

Total Kernel GPU Time (ms)

- Kernels Using TC
- Memory Kernels
- All Other Kernels

More...

Tensor Core Kernel Efficiency (%)

- Using TC
- Not Using TC

Performance Summary

Wall Clock Time (min)	2.17
Tensor Core Kernel Efficiency (%)	69.2
GPU Utilization (%)	16.7
Total Iterations	1
Profiled Iterations	1
Start Iteration	0
Stop Iteration	0
Average Iteration Time (min)	2.17

Iteration Summary

More...

Top 10 GPU Ops

GPU Time (ns)	Op Name	Op Type	Calls	TC Eligible	Using TC
2,492,911,136	/module/pretrain/train/train_step/no_grad::decorate_context/Float16OptimizerWithFloat16Params::step/wrapper/step	step	81	✗	✗
2,182,500,795	/module/pretrain/train/save_checkpoint_and_time/save_checkpoint/save	save	1	✗	✗
2,052,579,284	/module/pretrain/save_checkpoint/save	save	1	✗	✗
1,317,276,697	/Thread::_bootstrap/Thread::_bootstrap_inner/Thread:run/_pin_memory_loop/pin_memory/dictcomp/pin_memory/pin_memory	pin_memory	40	✓	✓
788,590,231	/module/pretrain/train/train_step/no_grad::decorate_context/Float16OptimizerWithFloat16Params::step/Float16OptimizerWithFloat16Params::_copy_model_grads_to_main_grads/float	float	29200	✗	✗
715,347,840	/module/pretrain/train/train_step/no_grad::decorate_context/Float16OptimizerWithFloat16Params::step/Float16OptimizerWithFloat16Params::clip_grad_norm/clip_grad_norm_fp32/MultiTensorApply::__call__(2)/multi_tensor_scale	multi_tensor_scale	81	✗	✗
673,714,167	/_VocabParallelCrossEntropyBackward::apply/backward/ParallelTransformer::custom_forward/ParallelTransformerLayer::_call_impl/ParallelTransformerLayer::forward/ParallelAttention::_call_impl/ParallelAttention::forward/ColumnParallelLinear::_call_impl/ColumnParallelLinear::forward/linear	linear	2015	✓	✓
617,281,886	/module/pretrain/train/train_step/no_grad::decorate_context/Float16OptimizerWithFloat16Params::step/Float16OptimizerWithFloat16Params::_copy_main_params_to_model_params/_multi_tensor_copy_this_to_that/copy_	copy_	23652	✗	✗
582,065,750	/_VocabParallelCrossEntropyBackward::apply/backward/ParallelTransformer::custom_forward/ParallelTransformerLayer::_call_impl/ParallelTransformerLayer::forward/ParallelMLP::_call_impl(2)/ParallelMLP::forward/ColumnParallelLinear::_call_impl/ColumnParallelLinear::forward/linear	linear	1616	✓	✓
573,974,613	/_VocabParallelCrossEntropyBackward::apply/backward/ParallelTransformer::custom_forward/ParallelTransformerLayer::_call_impl/ParallelTransformerLayer::forward/ParallelMLP::_call_impl(2)/ParallelMLP::forward/RowParallelLinear::_call_impl(2)/RowParallelLinear::forward/linear	linear	1720	✓	✓

System Config

Profile Name	gpt3_750m_second_run
GPU Count	1
GPU Name(s)	RTX A6000
CPU Model	AMD Ryzen Threadripper 3970X 32-Core Processor
GPU Driver Version	460.73.01
Framework	PyTorch 1.9.0a0+df837d0
CUDA Version	11.2

Recommendations

Problem	Recommendation
Unable to split profile into training iterations: key node not found	Specify key node by setting the --key_node argument
Slow debug APIs were enabled. When not debugging, these APIs can slow down execution of your model	Do not use the record_function decorator or context manager unless debugging.
GPU Memory is underutilized: Only 41% of GPU Memory is used	Try increasing batch size by 2x to increase data throughput

Guidance

Understanding GPU utilization and timing details of the operations is the first step in profiling your model.

- To learn more about Tensor cores and Mixed Precision training, visit this site:https://developer.nvidia.com/tensor_cores
- You will find resources on how to train networks with mixed precision and make full use of Tensor cores for Tensorflow models here: https://docs.nvidia.com/deeplearning/sdk/mixed-precision-training/index.html#training_tensorflow
- Note that if there are multiple kernels being observed on single op node, these are likely performing data transposes to prepare the data for efficient use by tensorcores. Such transposes themselves would not use tensor cores.

TensorBoard DLPROF

Profiling - 1st trial naive run

INACTIVE

GPU Utilization (%)

Total Op GPU Time (ms)

More...

Total Kernel GPU Time (ms)

More...

Tensor Core Kernel Efficiency (%)

System Config

Profile Name	gpt3_750m_naive
GPU Count	1
GPU Name(s)	RTX A6000
CPU Model	AMD Ryzen Threadripper 3970X 32-Core Processor
GPU Driver Version	460.73.01
Framework	PyTorch 1.9.0a0+df837d0
CUDA Version	11.2

Recommendations

Problem	Recommendation
61 ops were eligible to use tensor cores but none are using FP16	Try enabling AMP (Automatic Mixed Precision). For more information: https://developer.nvidia.com/automatic-mixed-precision
Unable to split profile into training iterations: key node not found	Specify key node by setting the --key_node argument
Slow debug APIs were enabled. When not debugging, these APIs can slow down execution of your model	Do not use the record_function decorator or context manager unless debugging.
GPU Memory is underutilized: Only 38% of GPU Memory is used	Try increasing batch size by 2x to increase data throughput

Guidance

Understanding GPU utilization and timing details of the operations is the first step in profiling your model.

- To learn more about Tensor cores and Mixed Precision training, visit this site:https://developer.nvidia.com/tensor_cores
- You will find resources on how to train networks with mixed precision and make full use of Tensor cores for Tensorflow models here: https://docs.nvidia.com/deeplearning/sdk/mixed-precision-training/index.html#training_tensorflow
- Note that if there are multiple kernels being observed on single op node, these are likely performing data transposes to prepare the data for efficient use by tensorcores. Such transposes themselves would not use tensor cores.

After applying **recommendation** from DLProf

TensorBoard DLPROF

Profiling - 2nd run applying recommended actions

INACTIVE

GPU Utilization (%)

Total Op GPU Time (ms)

More...

Total Kernel GPU Time (ms)

More...

Tensor Core Kernel Efficiency (%)