HPC.NRW

aiXcelerate 2021: Part II - Machine Learning (ML)

HPC.NRW Competence Network

Identifying whether my workload makes efficient use of resources

Jannis Klinkenberg (RWTH), Philipp Martin (RWTH), Radita Liem (RWTH), Anara Kozhokanova (RWTH)

aiXcelerate 2021: Part II - Machine Learning (ML)

Reviewing Schedule – Machine Learning (ML)

HPC.NRW

Wednesday, December 8

Start	End	Topic	Speaker
09:00	09:30	Welcome & ML Focus	RWTH
09:30	10:00	Infrastructure Overview and Future Plans	Christian Terboven (RWTH)
10:00	10:15	Break	
10:15	11:00	Vision: Development, Testing and Production	Jannis Klinkenberg (RWTH)
11:15	11:30	Break	
11:30	12:30	Configuring and running ML/DL workloads on CLAIX	Sven Hansen (RWTH), Jannis Klinkenberg (RWTH)
12:30	14:00	Lunch Break	
14:00	15:00	Identifying utilization efficiency	Jannis Klinkenberg (RWTH)
15:00	15:15	Break	
15:15	16:45	Hands-on: Running ML/DL Workloads on CLAIX	Jannis Klinkenberg (RWTH)

Thursday, December 9

Start	End	Topic	Speaker
09:00	10:30	BYO: Reviewing & adapting user codes and workflows	
10:30	10:45	Break	
10:45	12:15	BYO: Reviewing & adapting user codes and workflows	
12:15	13:45	Lunch Break	
13:45	14:45	BYO: Review results	
14:45	15:00	Break	
15:00	16:00	BYO: Lightning talks about take-aways	

Motivation

HPC.NRW

- **Question:** How do I know whether my application makes efficient use of resources and how do I identify issues or bottlenecks?
  - Example: Small or shallow DL model with only few data running on a large GPU
  - Try to avoid pathologic cases
  - Which tools are available?

Detailed Agenda for Session

HPC.NRW

- Basic utilization checks for GPU applications – Jannis Klinkenberg (RWTH)
- Teaser: TensorBoard – Jannis Klinkenberg (RWTH)
- ML / DL applications with heavy I/O – Philipp Martin (RWTH)
  - Bringing it all together: ML / DL + I/O
- Case Study: Performance analysis of large-scale distributed Deep Learning – Radita Liem (RWTH) and Anara Kozhokanova (RWTH)
  - What is possible in terms of distributed deep learning?
  - Where can we support users with their codes?

Basic Utilization Checks for GPU Applications

HPC.NRW

- **Reminder:** CLAIX GPU nodes feature Nvidia GPUs
  - Tesla V100 on Skylake partition (c18g)
  - Tesla P100 on Broadwell partition (c16g)
- **Idea:** Use Nvidia System Management Interface (`nvidia-smi`)
  - Provides monitoring and management capabilities
  - Information about
    - Configuration (e.g., compute mode)
    - Clock speed, power draw & temperature
    - Processes running on GPU
    - Utilization (GPU, memory, encoder / decoder)
  - Output can be configured and written to csv or stdout
    - e.g., filter what you want to see and in what interval

Basic Utilization Checks for GPU Applications

HPC.NRW

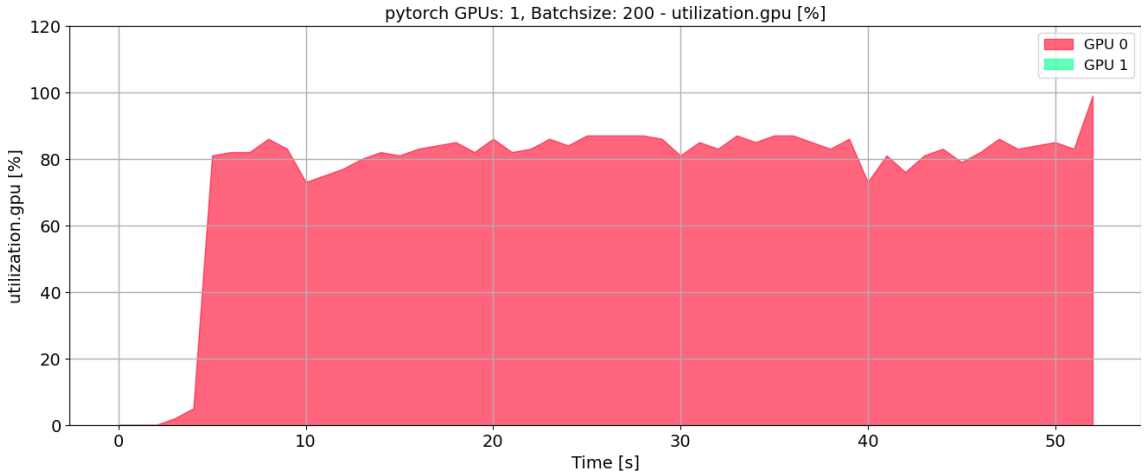
– **Approach:** Wrap your execution

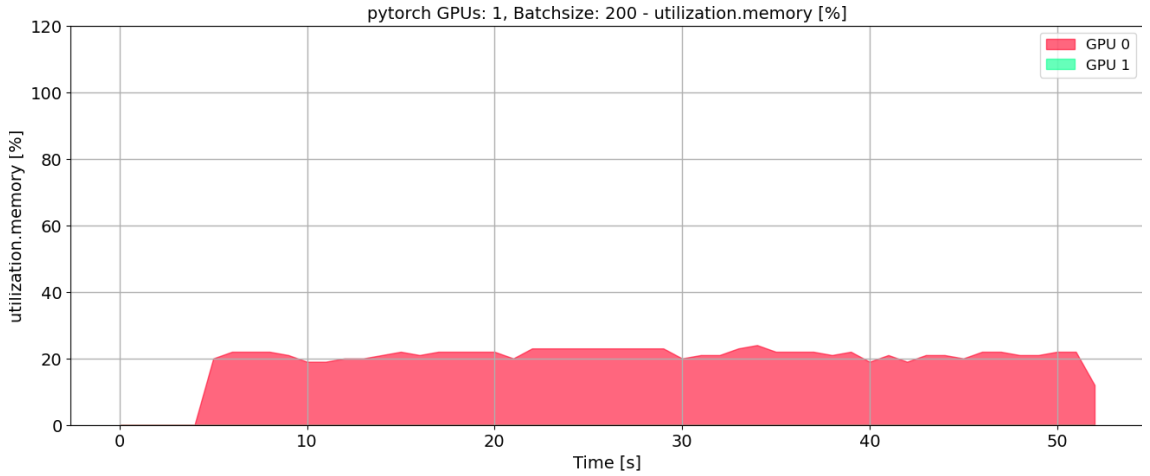
```
#!/bin/bash

# repeat query every two second and pipe result to separate file
nvidia-smi --query-gpu=timestamp,index,compute_mode,pstate,utilization.gpu,utilization.memory,
memory.used,temperature.gpu,power.draw --format=csv --loop=2 &> <file_name_monitoring>.txt &

# remember ID of process that has just been started in background
export proc_id_monitor=$!

# execute your program
<exec statement> <params>

# stop monitoring again
kill -2 ${proc_id_monitor}
sleep 5
```

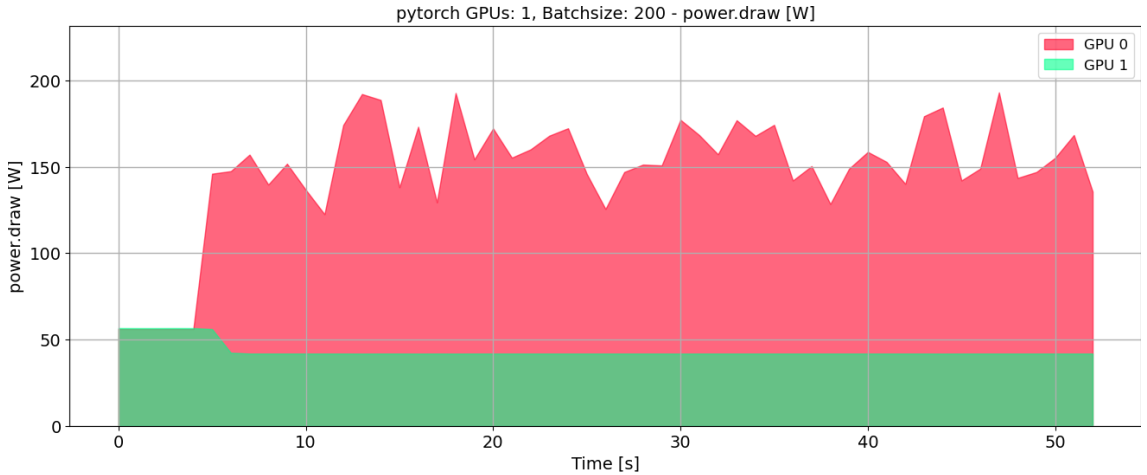
Result:

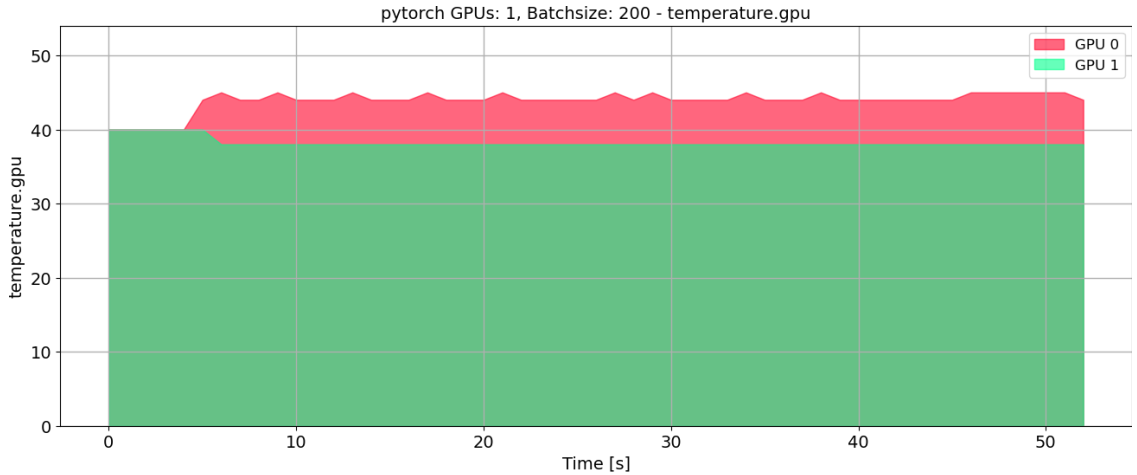
```
timestamp, index, compute_mode, pstate, utilization.gpu [%], utilization.memory [%], ...
2021/11/12 14:24:52.010, 1, Default, P0, 0 %, 0 %, 0 MiB, 40, 56.54 W
2021/11/12 14:24:54.012, 0, Default, P0, 81 %, 20 %, 2758 MiB, 44, 145.93 W
2021/11/12 14:24:54.013, 1, Default, P0, 0 %, 0 %, 0 MiB, 40, 56.01 W
2021/11/12 14:24:56.015, 0, Default, P0, 82 %, 22 %, 2758 MiB, 45, 147.40 W
2021/11/12 14:24:56.016, 1, Default, P0, 0 %, 0 %, 0 MiB, 38, 42.31 W
2021/11/12 14:24:58.017, 0, Default, P0, 82 %, 22 %, 2758 MiB, 44, 156.98 W
2021/11/12 14:24:58.019, 1, Default, P0, 0 %, 0 %, 0 MiB, 38, 41.82 W
2021/11/12 14:25:00.020, 0, Default, P0, 86 %, 22 %, 2758 MiB, 44, 139.66 W
2021/11/12 14:25:00.022, 1, Default, P0, 0 %, 0 %, 0 MiB, 38, 41.82 W
2021/11/12 14:25:02.023, 0, Default, P0, 83 %, 21 %, 2758 MiB, 45, 151.70 W
2021/11/12 14:25:02.024, 1, Default, P0, 0 %, 0 %, 0 MiB, 38, 41.82 W
2021/11/12 14:25:04.026, 0, Default, P0, 73 %, 19 %, 2758 MiB, 44, 136.63 W
...
```

Basic Utilization Checks for GPU Applications









Performance Profiling & Tracing for ML / DL Codes

HPC.NRW

- We have seen some tools already yesterday (Score-P, CUBE, Vampir)
- **Teaser:** TensorFlow Profiler (similar: Nsight Systems for PyTorch)
  - Create profile and trace of your application
  - Identify issues and receive recommendations
  - Details: **Live Session**

- Links:
  - https://www.tensorflow.org/tensorboard/tensorboard_profiling_keras
  - https://www.tensorflow.org/guide/profiler

ML / DL applications performing heavy I/O

HPC.NRW

- Refer to the I/O slides from the last two days
  - The tool “darshan” can be used to profile the I/O patterns of your application
  - In the hands-on PyTorch code, there is an example of how to do so
  - Darshan may need to be custom compiled, for example to work in a container
- ImageNet dataset
  - Large dataset, often used in machine learning research
  - Now available in a central location
  - Contact servicedesk@itc.rwth-aachen.de to get access (required due to licensing issues)

Running Darshan with a PyTorch Singularity container

HPC.NRW

- Installation of custom darshan
  - The PyTorch container does not supply certain libraries
  - Compile darshan-3.3.1 with `--disable-lustre-mod` and `--without-mpi`
  - Compile using gcc, not intel (`module unload intel intelmpi`)
- How to use
  - Set the `LD_PRELOAD` environment variable in the run script
  - Make sure to only do so for the command you want to analyze
  - Set `export DARSHAN_ENABLE_NONMPI=1`
  - Analyze the logs with `darshan-job-summary.pl` after loading LaTeX (`module load MISC texlive`)

Case Study: Performance Analysis of Large Scale Distributed Deep Learning