**Guidance to run PyTorch BERT-Large PreTraining on NVIDIA H100 GPUs**

Login to ACES cluster and run the commands below.

$cd $SCRATCH

$mkdir h100-benchmarks

$cd h100-benchmarks

$git clone -b core\_r0.4.0 <https://github.com/NVIDIA/Megatron-LM.git>

# Below change is required to print average throughput

# Update $SCRATCH/Megatron-LM/megatron/training.py with (utils/training.py)

# create a slurm job file test\_pytorch\_bert\_large.slurm and copy and paste the content below to it.

$vim test\_pytorch\_bert\_large.slurm

#!/bin/bash

##ESSARY JOB SPECIFICATIONS

#SBATCH --job-name=<your\_job\_name>

#SBATCH --time=1:00:00 #Set the wall clock limit to 5hr

#SBATCH --nodes=1

#SBATCH --ntasks=1

#SBATCH --cpus-per-task=8

#SBATCH --mem=80GB

#SBATCH --output=<your\_job>\_run.%j

#SBATCH --partition=gpu #Request 1 GPU per node can be 1 or

#SBATCH --gres=gpu:h100:1 #Request 1 GPU per node can be 1 or 2

#This command is used to get stats of H100 GPU utilization

nvidia-smi --query-gpu=timestamp,name,pci.bus\_id,driver\_version,pstate,pcie.link.gen.max,pcie.link.gen.current,temperature.gpu,utilization.gpu,utilization.memory,memory.total,memory.free,memory.used --format=csv -l 1 > <your\_job>\_GPU\_stats.log &

#This command is used to get stats of CPU cores utilization

watch -n 5 ps -u $USER > <your\_job>\_CPU\_stats.log &

export SINGULARITY\_BINDPATH="$SCRATCH/h100-benchmarks/Megatron-LM:/workspace,

/scratch/data/pytorch-language-modelling-datasets:/shared\_space\_datasets"

export CUDA\_DEVICE\_MAX\_CONNECTIONS=1

GPUS\_PER\_NODE=1

NNODES=1

WORLD\_SIZE=$(($GPUS\_PER\_NODE\*$NNODES))

NPROCS\_PER\_NODE=1

NPROCS=1

MASTER=`/bin/hostname -s`

SLAVES=`scontrol show hostnames $SLURM\_JOB\_NODELIST | grep -v $MASTER`

HOSTLIST="$MASTER $SLAVES"

echo $HOSTLIST

echo head node: $MASTER

VOCAB\_FILE=/shared\_space\_datasets/vocab.txt

DATA\_PATH=/shared\_space\_datasets/intel-bert\_text\_sentence

micro\_batch\_size=64

global\_batch\_size=64

train\_iters=5000

precision=bf16

BERT\_ARGS="

--num-layers 24 \

--hidden-size 1024 \

--num-attention-heads 16 \

--seq-length 512 \

--max-position-embeddings 512 \

--micro-batch-size ${micro\_batch\_size} \

--global-batch-size ${global\_batch\_size} \

--lr 0.0001 \

--train-iters ${train\_iters} \

--lr-decay-iters 990000 \

--lr-decay-style linear \

--min-lr 0.00001 \

--weight-decay 1e-2 \

--lr-warmup-fraction .01 \

--clip-grad 1.0 \

--${precision}

"

DATA\_ARGS="

--data-path $DATA\_PATH \

--vocab-file $VOCAB\_FILE \

--split 450,32,20

"

OUTPUT\_ARGS="

--log-interval 100 \

--save-interval 10000 \

--eval-interval 1000 \

--eval-iters 10

"

singularity exec --nv /scratch/data/containers/nvidia-containers/pytorch-nemo-23-06.sif torchrun --nnodes $NNODES \

--nproc\_per\_node $NPROCS\_PER\_NODE \

--rdzv\_id $RANDOM \

--rdzv\_backend c10d \

--rdzv\_endpoint $MASTER:${RANDOM} \

/workspace/pretrain\_bert.py \

$BERT\_ARGS \

$DATA\_ARGS \

$OUTPUT\_ARGS

$sbatch test\_pytorch\_bert\_large.slurm