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
```