OLCF Training

Overveiw of AI Stack on Frontier

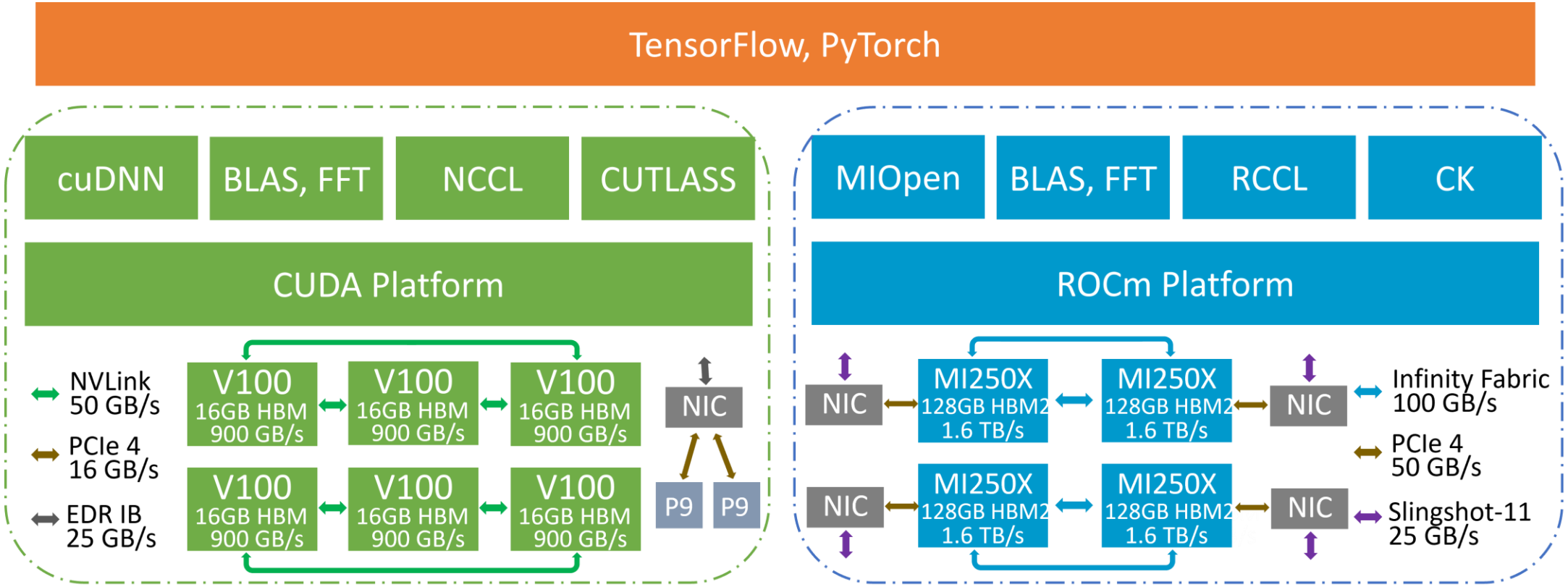
Junqi Yin

Analytics and AI Methods at Scale

Mar 28, 2025

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

OLCF Platforms: AI Stack



Summit – 200PFLOPS

Retired

Frontier – 1.6EFLOPS

Distributed Training Strategies on Frontier

- Data Parallel
  - PyTorch DDP, TensorFlow Multiworker Mirrored strategy
  - Horovod
- Model Parallel
  - PyTorch FSDP
  - DeepSpeed-Megatron 3D parallelism

Frameworks built-in

Third party

AI Software Environment

- PyTorch and TensorFlow are pip installable: PyTorch on Frontier

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh -b -p $PWD/miniconda3
export PATH=$PWD/miniconda3/bin:$PATH
conda create --prefix $PWD/miniconda3/envs/my_env
source $PWD/miniconda3/etc/profile.d/conda.sh
conda activate $PWD/miniconda3/envs/my_env

pip install torch==2.4.0 torchvision==0.19.0 torchaudio==2.4.0 --index-url https://download.pytorch.org/whl/rocm6.1

DS_BUILD_FUSED_LAMB=1 pip install deepspeed == 0.15.1

git clone https://github.com/mpi4py/mpi4py.git
cd mpi4py
CC=$(which mpicc) CXX=$(which mpicxx) python setup.py build --mpicc=$(which mpicc)
CC=$(which mpicc) CXX=$(which mpicxx) python setup.py install

git clone https://github.com/ROCm/apex.git
cd apex
pip install -r requirements.txt
pip install -v --no-build-isolation --config-settings "--build-option=--cpp_ext" --config-settings "--build-option=--cuda_ext" ./
```

AI Software Environment

- Install RCCL plugin: PyTorch on Frontier

```
git clone https://github.com/ROCm/aws-ofi-rccl
CC=cc ./configure --with-libfabric=/opt/cray/libfabric/1.22.0 --with-hip=/opt/rocm-6.1.3 --with-rccl=/opt/rocm-6.1.3/rccl --
prefix=$PWD/miniconda3/lib --enable-trace
```

- PyTorch distributed initialization

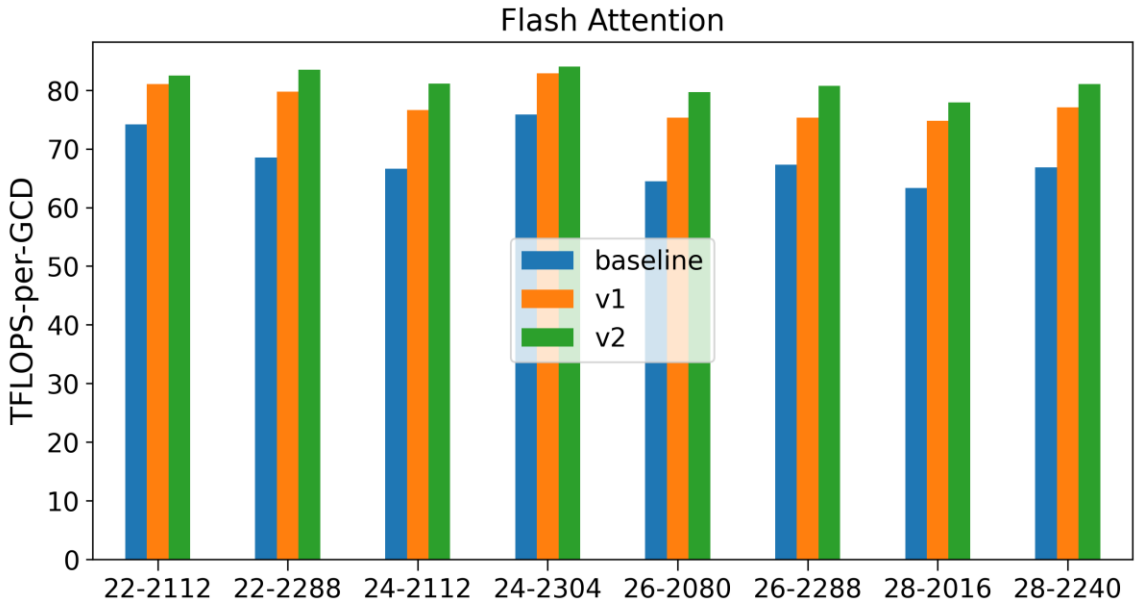
```
os.environ['RANK'] = os.environ['SLURM_PROCID']
os.environ['WORLD_SIZE'] = os.environ['SLURM_NTASKS']
os.environ['LOCAL_RANK'] = os.environ['SLURM_LOCALID']
dist.init_process_group(
        backend="nccl",
        init_method='env://',
        rank=rank,
        world_size=world_size)
```

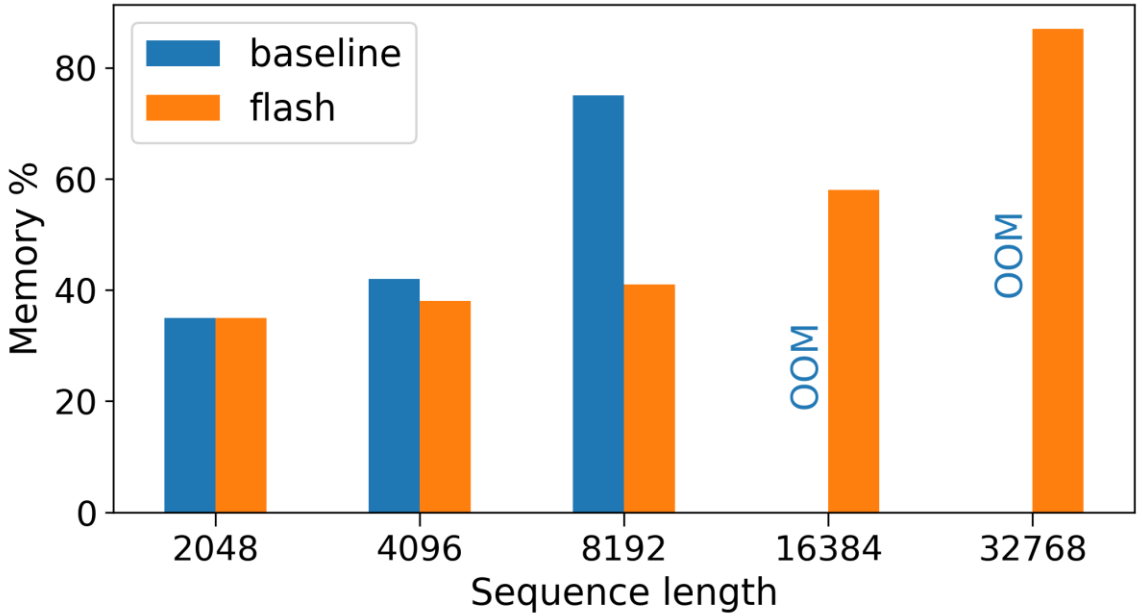
- Launching distributed job

```
export MASTER_ADDR=$(hostname -i)
export MASTER_PORT=29500
srun -N$SLURM_JOB_NUM_NODES -n$((SLURM_JOB_NUM_NODES*8)) -c7 --gpus-per-task=1 --gpu-bind=closest python train.py
```

AI scaling practice

- Flash attention (V2 supported by MI250X)
  - ~20% boost
  - 4X longer sequence
- PyTorch scaled_dot_product_attention

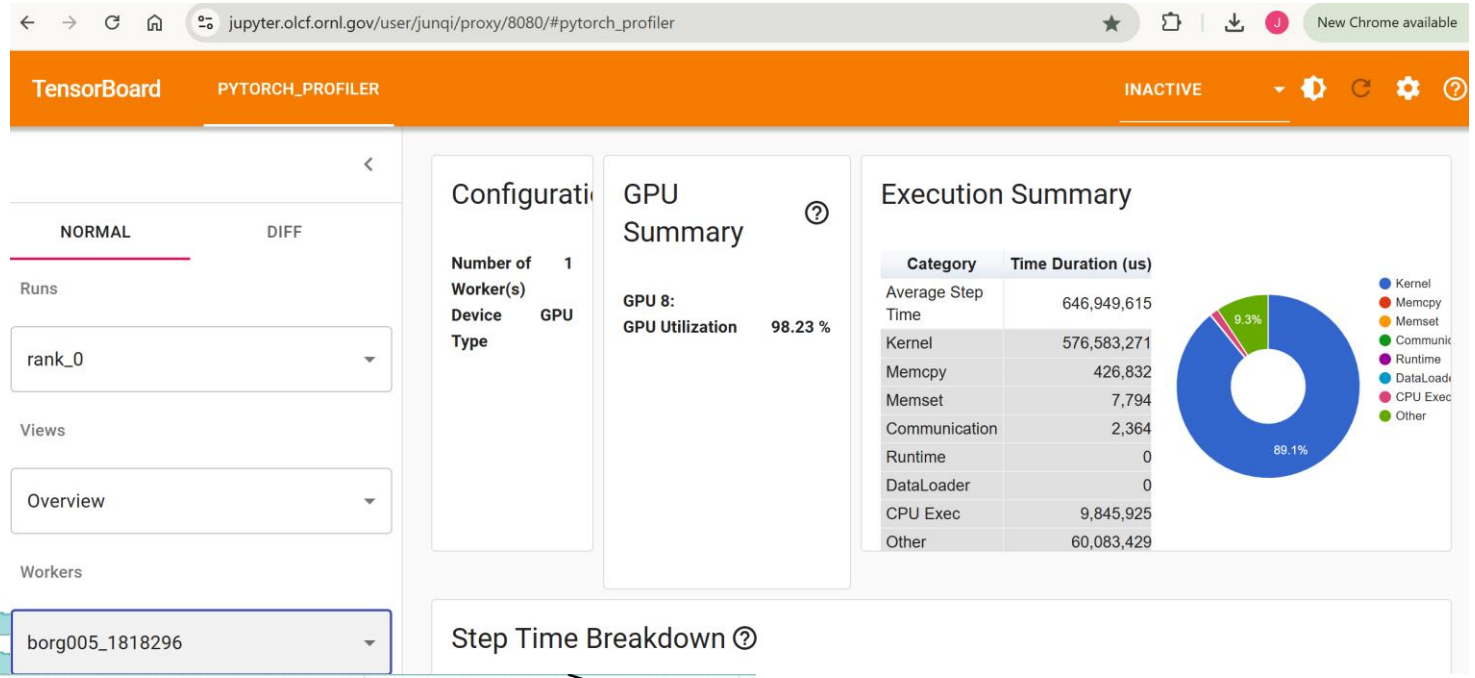


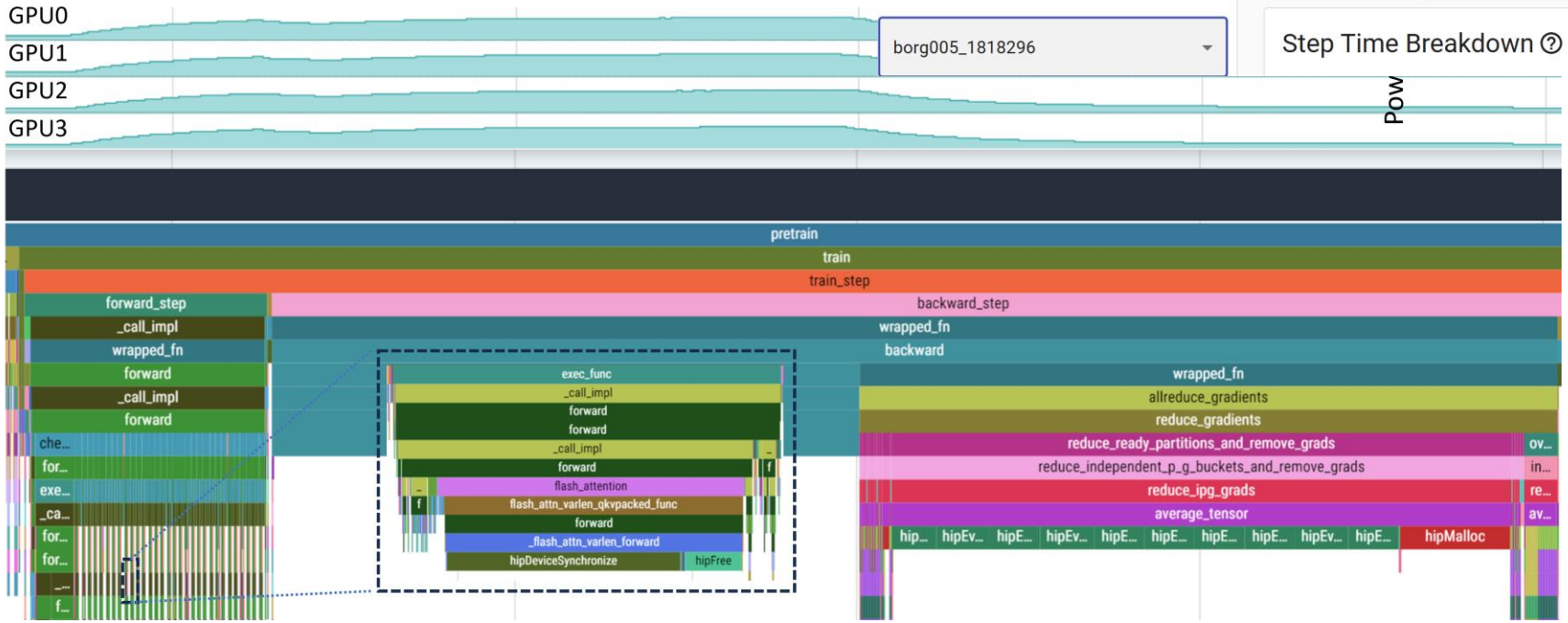


Comparative study of LLMs on Frontier

AI scaling practice

- Profiling tools
  - PyTorch profiler
  - Omnitrace





OAK RIDGE National Laboratory

IO considerations

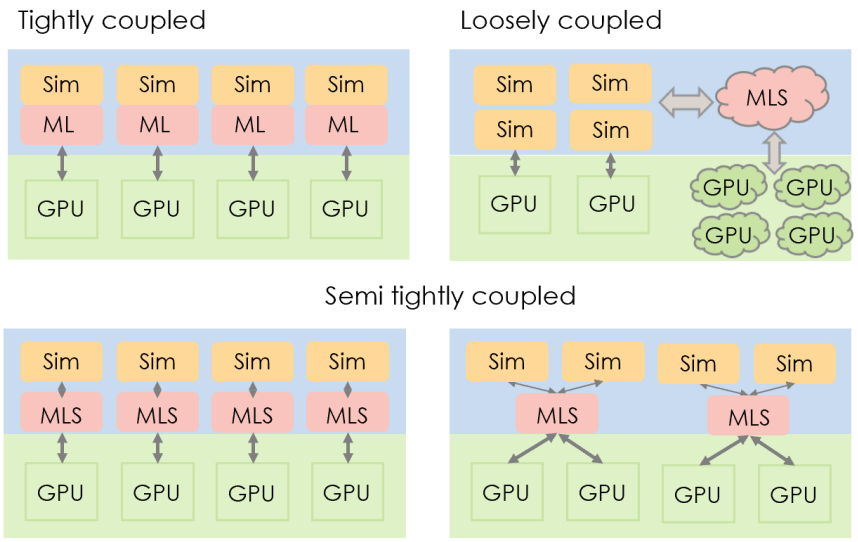
- For many-small-files data, considering optimized data format, such as LMDB
- Use multiprocessing and prefetching to hide the IO overhead

```
dataloader = DataLoader(dataset,
                        num_workers=7,
                        prefetch_factor=2,
                        pin_memory=torch.cuda.is_available())
```

- PyTorch Data Loading
- Frontier NVMe best practice

Interleaved Mod-Sim Surrogate

- Integration methods and features



Method	Pro	Con
Framework C++ API (TensorFlow/PyTorch C++)	• Portable • Better latency • Easy to deploy	• Not flexible
Framework Server (TensorFlow Serving/TorchServe)	• Flexible • Better throughput • Options to deploy	• High maintenance
Third-party API (SmartRedis/RedisAI)	• Easy integration • More functionality • Model support	• Portability

- SmartSim on Frontier

Strategies for Integrating Simulation with AI Surrogate on Frontier