Michael's Tip Corner

- PyTorch on Frontier docs have been updated!
  - https://docs.olcf.ornl.gov/software/analytics/pytorch_frontier.html
- Includes a new benchmarking example, FlashAttention builds instructions, Torch 2.8+ROCm 6.4.1 recommendation

Example Usage

We adapted the `multinode.py` DDP tutorial and simplified AMD's microbenchmarking script to work with SLURM, `mpi4py`, and to use 1 GPU per MPI task. Utilizing all the GPUs on the node in this manner means there will be 8 tasks per node. Because we are enforcing 1 GPU per task, each MPI task only sees device `0` in PyTorch. Even if the *physical* GPU ID on Frontier is different, and even though there are 8 GCDs (GPUs) on a node, **the torch device in this case is still 0** due to a task only being mapped to one GPU.

Both scripts below use `DistributedDataParallel` and can run across multiple nodes.

multinode_olcf.py >

microbench_olcf.py >

To run the python scripts, an example batch script is given below:

Batch Script >

Flash Attention

In addition to PyTorch's internal implementation of FlashAttention, som FlashAttention. To install the `flash-attn` library on Frontier:

```
# Activate your virtual environment
source activate /path/to/my_env

# Install some build tools
pip install ninja packaging

# Retrieve the FA repo
git clone https://github.com/ROCm/flash-attention
cd flash-attention/
git checkout v2.7.4-cktile
git submodule init
git submodule update

# Build the flash-attn wheel
python3 setup.py bdist_wheel

# Install flash-attn
pip install dist/*.whl
```