Parallel Communication on Heterogeneous Architecture

Amanda Bienz
William D. Gropp
Luke Olson

University of Illinois at Urbana-Champaign
CPU vs GPU

**CPU**
- A small number of complex cores
- Referred to as ‘Host’ throughout presentation
- HAL: 20 cores per CPU

**GPU**
- Large number of simple cores
- Floating point operations are very cheap
- Referred to as ‘Device’ throughout presentation
- HAL: 5120 cores per GPU
Neural Networks on HAL

- Standard Pytorch
  - Accelerate neural network with GPU
- Data Parallel Mode
  - Accelerate with multiple GPUs
- Distributed
  - Use multiple nodes with MPI

```python
import torch
import torch.nn as nn

class DataParallelModel(nn.Module):
    def __init__(self):
        super().__init__()
        self.block1 = nn.Linear(10, 20)

        # wrap block2 in DataParallel
        self.block2 = nn.Linear(20, 20)
        self.block2 = nn.DataParallel(self.block2)

        self.block3 = nn.Linear(20, 10)

    def forward(self, x):
        #!/usr/bin/env python
        x = self.block1
        x = self.block2
        return x

import torch
import os

import torch.distributed as dist

def run(rank, size):
    #*** Distributed function to be implemented later
    pass

def init_processes(rank, size, fn, backend='tcp'):
    #*** Initialize the distributed environment.***
    os.environ['MASTER_ADDR'] = '127.0.0.1'
    os.environ['MASTER_PORT'] = '29500'
    dist.init_process_group(backend, rank=rank, world_size=size)
```

pytorch.org
What parts of code should be on GPU?

How should code be distributed across many GPUs/nodes?

How should data be transferred between CPUs and GPUs?

How should data be transported between nodes?
Scale Performance

Scaling floating point values on CPU or GPU

Time to perform scale operation

Performance of scale operation
Scale Performance

### SPECIFICATIONS

<table>
<thead>
<tr>
<th></th>
<th>Tesla V100 PCIe</th>
<th>Tesla V100 SXM2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GPU Architecture</strong></td>
<td>NVIDIA Volta</td>
<td></td>
</tr>
<tr>
<td><strong>NVIDIA Tensor Cores</strong></td>
<td>640</td>
<td></td>
</tr>
<tr>
<td><strong>NVIDIA CUDA® Cores</strong></td>
<td>5,120</td>
<td></td>
</tr>
<tr>
<td><strong>Double-Precision</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Single-Precision</strong></td>
<td><strong>14 TFLOPS</strong></td>
<td><strong>15.7 TFLOPS</strong></td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GPU Memory</strong></td>
<td>32GB /16GB HBM2</td>
<td></td>
</tr>
<tr>
<td><strong>Memory Bandwidth</strong></td>
<td>900GB/sec</td>
<td></td>
</tr>
<tr>
<td><strong>ECC</strong></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td><strong>Interconnect Bandwidth</strong></td>
<td>32GB/sec</td>
<td>300GB/sec</td>
</tr>
<tr>
<td><strong>System Interface</strong></td>
<td>PCIe Gen3</td>
<td>NVIDIA NVLink</td>
</tr>
<tr>
<td><strong>Form Factor</strong></td>
<td>PCIe Full Height/Length</td>
<td>SXM2</td>
</tr>
<tr>
<td><strong>Max Power Consumption</strong></td>
<td>250 W</td>
<td>300 W</td>
</tr>
<tr>
<td><strong>Thermal Solution</strong></td>
<td>Passive</td>
<td></td>
</tr>
<tr>
<td><strong>Compute APIs</strong></td>
<td>CUDA, DirectCompute, OpenCL™, OpenACC</td>
<td></td>
</tr>
</tbody>
</table>

Performance of scale operation on CPU or GPU

![Graph showing performance comparison between Host and Device](image-url)
Data Movement

1. Data starts here

2. Move to GPU

3. Do operation on GPU

4. Move results back to CPU
Memcpy Between CPU and GPU

Moving data between the CPU and GPU

Time to move data

Performance of data movement
Single Node on HAL

GPU 0  GPU 1
   |     |   |
CPU 0  |     | CPU 1
     |     |   |
GPU 2  GPU 3

Network
Comparison with GPU to GPU

Moving data between the GPUs

Time to move data

Performance of data movement
Memcpy GPU to GPU

524288 Bytes, Not Transferred To Self (Diagonal)
Two Nodes on HAL
Cuda-Aware MPI

- Compile mpi program with spectrum-mpi
  - module unload openmpi
  - module load spectrummpi

```bash
CXX_FLAGS=-I/opt/ibm/spectrum_mpi/include -L/opt/ibm/spectrum_mpi/lib -lmpi_ibm
nvcc ${CXX_FLAGS} <filename>
```

- In runscript, get hostfile from srun

```bash
srun hostname > hosts
mpirun -np 2 --map-by node --hostfile hosts -gpu
```
MPI Ping-Pong

Sending data between 2 nodes
1 process per node

Time to move data

Performance of data movement
Ping-Pong GPU to GPU

524288 Bytes; Between 2 Nodes, 1 PPN