Parallelization of Large Scale Deep Neural Networks

IISc-Shell Work Progress Meet
(15/05/2024)
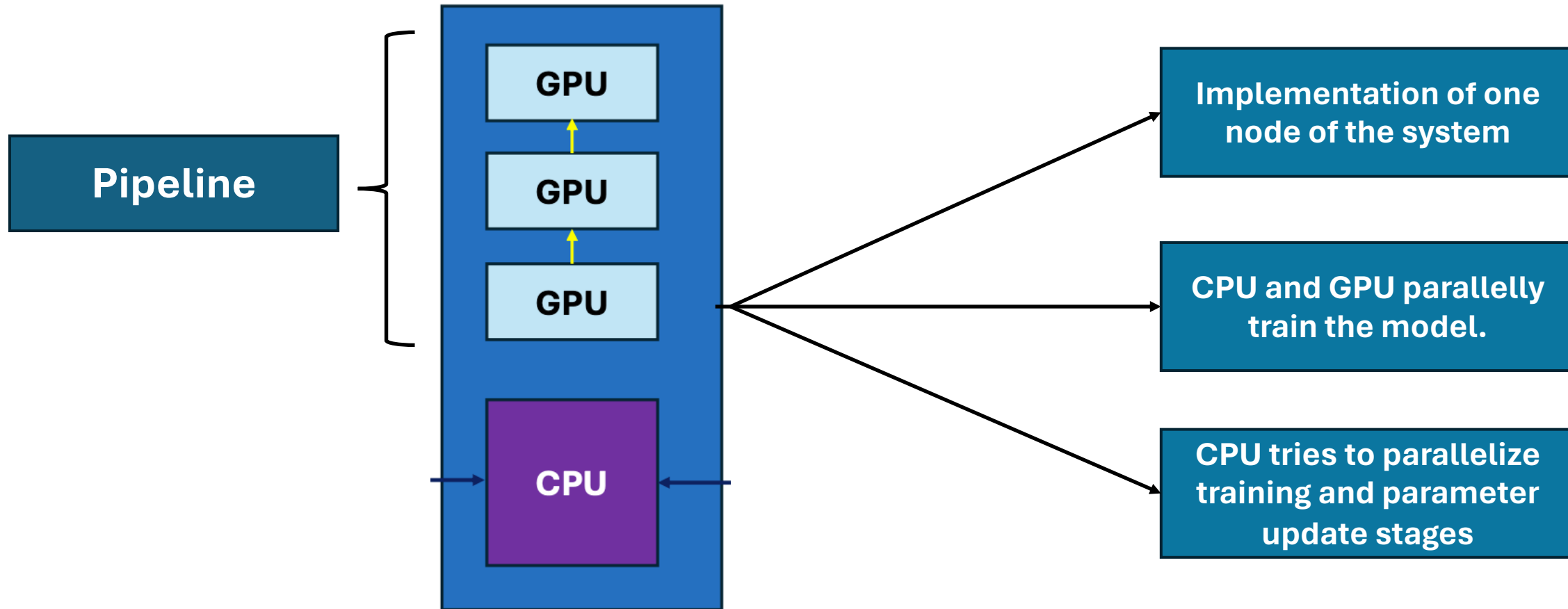
Dheemanth R Joshi
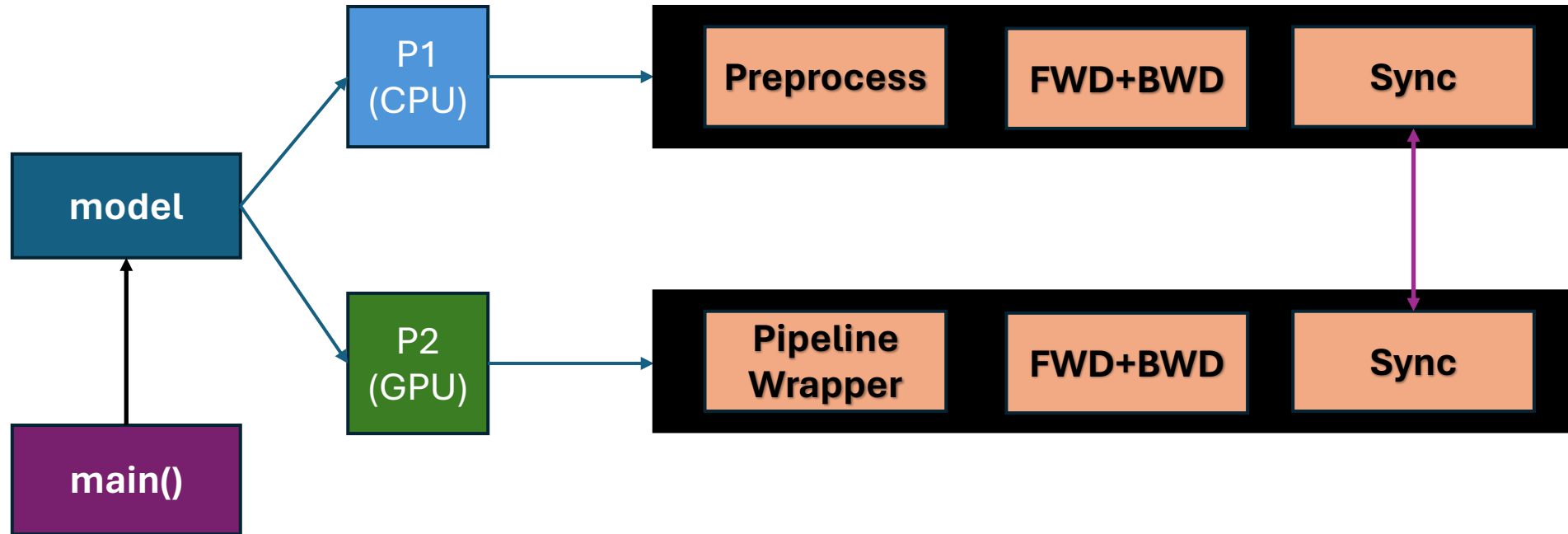
Intern

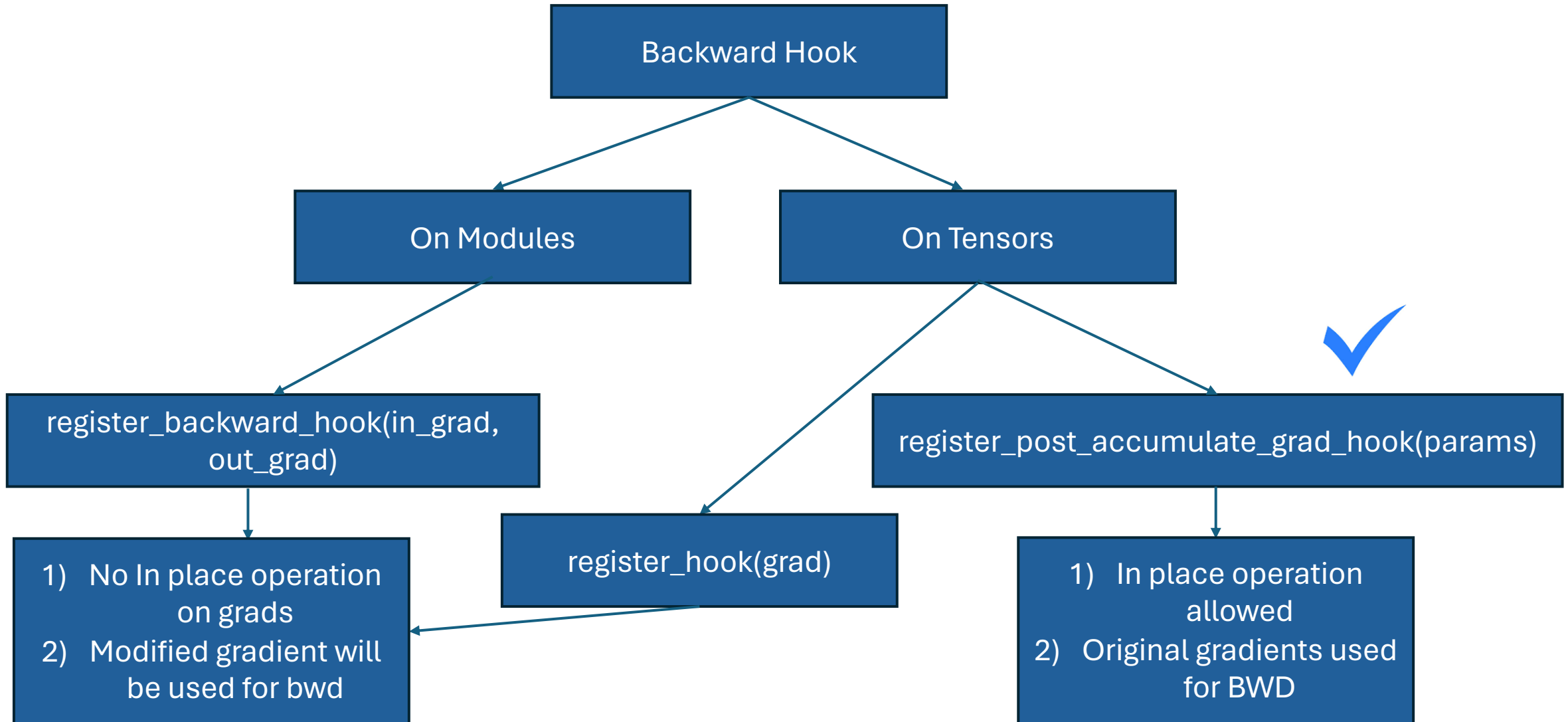**MARS Lab, CDS, IISc**

Ongoing Work

Implementation of 1 Node

Current Implementation Status

hooks to initiate all-reduce

```
Backward Hook
```

```
On Modules
```

```
On Tensors
```

```
register_backward_hook(in_grad,
out_grad)
```

```
register_post_accumulate_grad_hook(params)
```

```
register_hook(grad)
```

```
1)  No In place operation
    on grads
2)  Modified gradient will
    be used for bwd
```

```
1)  In place operation
    allowed
2)  Original gradients used
    for BWD
```

Code Walkthrough

Loss Convergence

```
[2024-05-14 03:07:03,527] torch.distributed.run: [WARNING]
[2024-05-14 03:07:03,527] torch.distributed.run: [WARNING] ****************************************
[2024-05-14 03:07:03,527] torch.distributed.run: [WARNING] Setting OMP_NUM_THREADS environment variable for each process to be 1 in default, to avoid your system being overloaded, p
lease further tune the variable for optimal performance in your application as needed.
[2024-05-14 03:07:03,527] torch.distributed.run: [WARNING] ****************************************
/home/dheemanth/ishell/lib/python3.10/site-packages/torchgpipe/stream.py:99: UserWarning: TypedStorage is deprecated. It will be removed in the future and UntypedStorage will be the
 only storage class. This should only matter to you if you are using storages directly.  To access UntypedStorage directly, use tensor.untyped_storage() instead of tensor.storage()
  tensor = tensor.new_empty([0]).set_(tensor.storage())
Process 1 || Loss = 0.10652325302362442
Process 0 || Loss = 0.33383384346961975
Process 1 || Loss = 0.06168791651725769
Process 0 || Loss = 0.25596049427986145
Process 1 || Loss = 0.0458226352930069
Process 0 || Loss = 0.19206832349300385
Process 1 || Loss = 0.04945692792534828
Process 0 || Loss = 0.15888382494449615
Process 1 || Loss = 0.047659020870923996
Process 0 || Loss = 0.12711946666240692
Process 1 || Loss = 0.04387592896819115
Process 0 || Loss = 0.11065974086523056
Process 1 || Loss = 0.03350578621029854
Process 0 || Loss = 0.07034637778997421
Process 1 || Loss = 0.031545259058475494
Process 0 || Loss = 0.05479532107710838
Process 1 || Loss = 0.02318580262362957
Process 0 || Loss = 0.037631630897521970
Process 1 || Loss = 0.015959901735186577
Process 0 || Loss = 0.025134941563010216
Process 1 || Loss = 0.022328471764922142
Process 0 || Loss = 0.019330212846398354
Process 1 || Loss = 0.010770427994430065
Process 0 || Loss = 0.009839644655585289
Process 1 || Loss = 0.008287331089377403
Process 0 || Loss = 0.01403298694640398
Process 1 || Loss = 0.005160300061106682
Process 0 || Loss = 0.0153941204771399
Process 1 || Loss = 0.001780155929736793
Process 0 || Loss = 0.0013132636668160558
Process 1 || Loss = 0.0006979411118663847
Process 0 || Loss = 0.0017395681934431195
```

Immediate Future Work

- Test the strategy with popular GPT/BERT models.

- Run profiling to study the hardware utilization.

- Try to improvise the synchronization strategy.