PiPPy: Automated Pipeline Parallelism for PyTorch

PiPPy is a library that:

1) Automatically splits a model in pipeline without code changes

2) Distributes stages and weave connections for user

3) Orchestrates micro-batches with overlap

Why pipelining

1) Pillar of 3D-Parallel larger models

2) Hides communication latency

Using PiPPy in Four Steps

# Create model as on a single process
mod = MyModel()

# Create a pipe by splitting the model
pipe = Pipe.from_tracing(mod, ...)

# Create a driver to drive the distributed pipeline
dist_mod = PipelineDriver(pipe, chunks, ..., world_size, ...)

# Pass input to the distributed model
output = dist_mod(input)

PiPPy: github.com/pytorch/tau

Compiler-based Model Split

Pipe = Pipe.from_tracing(module)

T5 Example

Distributed Pipeline Runtime

PiPPy's PipelineDriver

Supported schedules:
- Fill Drain
- 1 Fwd 1 Bwd (1F1B)
- Interleaved 1F1B

Visualization

Results

<table>
<thead>
<tr>
<th># GPUs</th>
<th>DDP size</th>
<th>Examples/s</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>4</td>
<td>18.2</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>8</td>
<td>27.8</td>
<td>1.5x</td>
</tr>
<tr>
<td>128</td>
<td>16</td>
<td>53.8</td>
<td>3.0x</td>
</tr>
</tbody>
</table>