Initial PyTorch DDP results on Habana and ThetaGPU report:

Ran my usual unet setup with DDP orchestration. Did not enable power profiling metrics for this, as we know their trends from a few test runs and they were not important to the tests we actually ended up doing, while they would have added a lot of overhead.

Most model versions were run on both ThetaGPU and Habana. All of the ThetaGPU runs were run in interactive sessions of one full node, the Habana ones in my (André Evard) and Bruce Wilson’s Habana environments and a nearly fresh PyTorch environment with scikit-image installed. Internal runtime metrics were gathered, but the shell command ‘time’ was the primarily profiler in this instance, as a pure runtime metric sufficed here.

For recreation, thee majority of the commands used to run these unet trials can be found in the repo’s unet\_bench/run-commands.txt.

One example: time mpirun -n 2 --rank-by core $PYTHON unet.py --hpu --use\_lazy\_mode --distributed --run-name 64-test-cards-2-5 --epochs 5 --image-size 64 --batch-size 64 --cache-path kaggle\_cache --weights-file 64-t-c-2.pt --world-size 2 --num-workers 5

Until otherwise specified, the default scale is 64, and batch size is 64. Epochs was always 5.

Initial test, fanning out the number of workers (in dataloaders) on two cards:

(Habana)

* 1 worker:
  + total\_train\_time = 55.83728361129761 for 5 epochs
  + total\_eval\_time = 17.906968593597412 for 5 epochs
  + real 1m22.022s
  + user 4m30.289s
  + sys 0m30.866s
* 2 workers:
  + total\_train\_time = 70.28326392173767 for 5 epochs
  + total\_eval\_time = 32.59425926208496 for 5 epochs
  + real 1m50.763s
  + user 6m5.934s
  + sys 0m45.028s
* 4 workers:
* 6 workers:
  + total\_train\_time = 133.79579830169678 for 5 epochs
  + total\_eval\_time = 92.13056659698486 for 5 epochs
  + real 3m54.662s
  + user 13m5.610s
  + sys 1m35.592s
* 10 workers:
  + total\_train\_time = 201.89221811294556 for 5 epochs
  + total\_eval\_time = 149.71976399421692 for 5 epochs
  + real 6m0.727s
  + user 20m14.274s
  + sys 2m26.898s

(ThetaGPU)

* 1 worker:
  + total\_train\_time = 42.773289918899536 for 5 epochs
  + total\_eval\_time = 34.69579553604126 for 5 epochs
  + real 1m23.527s
  + user 3m22.571s
  + sys 0m32.603s
* 2 workers:
  + total\_train\_time = 45.85600280761719 for 5 epochs
  + total\_eval\_time = 34.29944109916687 for 5 epochs
  + real 1m26.730s
  + user 3m20.757s
  + sys 0m32.859s
* 4 workers:
  + total\_train\_time = 79.60046982765198 for 5 epochs
  + total\_eval\_time = 64.38306474685669 for 5 epochs
  + real 2m31.003s
  + user 5m35.700s
  + sys 0m58.461s
* 6 workers:
  + total\_train\_time = 107.11616635322571 for 5 epochs
  + total\_eval\_time = 95.25253367424011 for 5 epochs
  + real 3m31.594s
  + user 8m5.669s
  + sys 1m24.478s
* 10 workers:
  + total\_train\_time = 174.23038792610168 for 5 epochs
  + total\_eval\_time = 157.4055826663971 for 5 epochs
  + real 5m38.262s
  + user 12m44.122s
  + sys 2m15.754s

Clearly time scales quite severely in regards to number of workers here. This trend was observed with every run, both on Habana and ThetaGPU. Somewhat better on ThetaGPU, but hereafter num-workers was not directly tested.

Did a single image size 258 batch size 64 2 card run.

total\_train\_time = 690.3633544445038 for 5 epochs

total\_eval\_time = 63.609962940216064 for 5 epochs

real 12m52.353s

user 40m52.852s

sys 9m23.301s

Was going to test more, but batch size 128 crashed, and since Habana was down for maintenance afterwords I prioritized ThetaGPU.

For reference, this is the default single card run on Theta:

total\_train\_time = 33.765321493148804 for 5 epochs

total\_eval\_time = 0.23828840255737305 for 5 epochs

real 0m40.219s

user 61m46.320s

sys 0m16.184s

After fanning out the num workers tests, I tested number of cards with image size 256 and batch size 64:

* 1 card 1 worker:
  + total\_train\_time = 501.9254755973816 for 5 epochs
  + total\_eval\_time = 3.402388572692871 for 5 epochs
  + real 8m34.290s
  + user 284m28.557s
  + sys 1m57.123s
* 2 cards 2 workers:
  + total\_train\_time = 320.25806999206543 for 5 epochs
  + total\_eval\_time = 46.062544107437134 for 5 epochs
  + real 6m31.071s
  + user 15m49.227s
  + sys 4m2.786s
* 4 cards 4 workers:
  + total\_train\_time = 275.23619318008423 for 5 epochs
  + total\_eval\_time = 60.528496503829956 for 5 epochs
  + real 5m46.863s
  + user 24m28.712s
  + sys 31m13.897s
* 8 cards 8 workers:
  + total\_train\_time = 567.9980697631836 for 5 epochs
  + total\_eval\_time = 126.94650268554688 for 5 epochs
  + real 11m49.687s
  + user 72m10.332s
  + sys 131m43.286s

So on ThetaGPU, the middling number of cards did actually improve performance for the model on this scale. 8 was too many and weighed down in coordination overhead (presumably), however. This test was not duplicated on Habana for time constraints, but a similar trend can be observed in this last test.

Using the duplicated dataset (4x as many images, redundant images). Batch size 64, num workers = num cards.

* Theta, 1 card
  + total\_train\_time = 647.3309409618378 for 5 epochs
  + total\_eval\_time = 2.05793833732605 for 5 epochs
  + real 11m7.691s
  + user 985m28.045s
  + sys 1m14.717s
* Theta, 2 cards
  + total\_train\_time = 344.72846364974976 for 5 epochs
  + total\_eval\_time = 40.020914793014526 for 5 epochs
  + Real 6m37.675s
  + user 17m48.840s
  + sys 3m35.094s
* Theta, 4 cards
  + total\_train\_time = 274.3716642856598 for 5 epochs
  + total\_eval\_time = 50.66974329948425 for 5 epochs
  + real 5m40.915s
  + user 25m5.889s
  + sys 29m49.019s
* Theta, 8 cards
  + total\_train\_time = 539.2973132133484 for 5 epochs
  + total\_eval\_time = 106.1164972782135 for 5 epochs
  + real 12m38.670s
  + user 74m48.205s
  + sys 127m19.537s
* Habana, 1 card
  + total\_train\_time = 595.8973989486694 for 5 epochs
  + total\_eval\_time = 7.450488328933716 for 5 epochs
  + real 10m31.974s
  + user 36m7.410s
  + sys 2m5.358s
* Habana, 2 cards
  + total\_train\_time = 691.71950507164 for 5 epochs
  + total\_eval\_time = 56.025211334228516 for 5 epochs
  + real 12m48.828s
  + user 40m0.883s
  + sys 9m32.902s

Unfortunately I could not scale the Habana unet further upward, as the attempts to acquire 4 and 8 cards both crashed. I am unsure why, perhaps to do with the update that was done? Nonetheless, based on the 1 and 2 card performance we can estimate their performance would have only worsened, while the 2 and 4 card on ThetaGPU performed better as expected by the previous trial.