Distributed Data Parallel Benchmark

This tool is used to measure distributed training iteration time. This is helpful for evaluating the performance impact of code changes to `torch.nn.parallel.DistributedDataParallel`, `torch.distributed`, or anything in between.

It optionally produces a JSON file with all measurements, allowing for an easy A/B comparison of code, configuration, or environment. This comparison can be produced by `diff.py`.

Requirements

This benchmark depends on PyTorch and torchvision.

How to run

Run as many copies of this script as you have model replicas.

If you launch a single task per machine with multiple GPUs, consider using `torch.distributed.launch` to spawn multiple processes per machine.

Example output (only on rank 0):

```
-----------------------------------
PyTorch distributed benchmark suite
-----------------------------------

* PyTorch version: 1.4.0a0+05140f0
* CUDA version: 10.0
* Distributed backend: nccl

--- nvidia-smi topo -m ---

        GPU0    GPU1    GPU2    GPU3    GPU4    GPU5    GPU6    GPU7    mlx5_2  mlx5_0  mlx5
GPU0     X      NV1     NV1     NV2     NV2     SYS     SYS     SYS     SYS     PIX     SYS
GPU1    NV1      X      NV2     NV1     SYS     NV2     SYS     SYS     SYS     PIX     SYS
GPU2    NV1     NV2      X      NV2     SYS     SYS     NV1     SYS     SYS     PHB     SYS
GPU3    NV2     NV1     NV2      X      SYS     SYS     SYS     NV1     SYS     PHB     SYS
GPU4    NV2     SYS     SYS     SYS      X      NV1     NV1     NV2     PIX     SYS     PHB
GPU5    SYS     NV2     SYS     SYS     NV1      X      NV2     NV1     PIX     SYS     PHB
GPU6    SYS     SYS     NV1     SYS     NV1     NV2      X      NV2     PHB     SYS     PIX
GPU7    SYS     SYS     SYS     NV1     NV2     NV1     NV2      X      PHB     SYS     PIX
mlx5_2  SYS     SYS     SYS     SYS     PIX     PIX     PHB     PHB      X      SYS     PHB
mlx5_0  PIX     PIX     PHB     PHB     SYS     SYS     SYS     SYS     SYS      X      SYS
mlx5_3  SYS     SYS     SYS     SYS     PHB     PHB     PIX     PIX     PHB     SYS      X
mlx5_1  PHB     PHB     PIX     PIX     SYS     SYS     SYS     SYS     SYS     PHB     SYS
```

```
Legend:

  X    = Self
  SYS  = Connection traversing PCIe as well as the SMP interconnect between NUMA nodes (e.g.
  NODE = Connection traversing PCIe as well as the interconnect between PCIe Host Bridges wi
  PHB  = Connection traversing PCIe as well as a PCIe Host Bridge (typically the CPU)
  PXB  = Connection traversing multiple PCIe switches (without traversing the PCIe Host Bric
  PIX  = Connection traversing a single PCIe switch
  NV#  = Connection traversing a bonded set of # NVLinks

--------------------------



Benchmark: resnet50 with batch size 32

                            sec/iter    ex/sec      sec/iter    ex/sec      sec/iter    ex/s
   1 GPUs --   no ddp:  p50:  0.097s     329/s  p75:  0.097s     329/s  p90:  0.097s     329
   1 GPUs --    1M/1G:  p50:  0.100s     319/s  p75:  0.100s     318/s  p90:  0.100s     318
   2 GPUs --    1M/2G:  p50:  0.103s     310/s  p75:  0.103s     310/s  p90:  0.103s     310
   4 GPUs --    1M/4G:  p50:  0.103s     310/s  p75:  0.103s     310/s  p90:  0.103s     310
   8 GPUs --    1M/8G:  p50:  0.104s     307/s  p75:  0.104s     307/s  p90:  0.104s     306
  16 GPUs --    2M/8G:  p50:  0.104s     306/s  p75:  0.104s     306/s  p90:  0.104s     306

Benchmark: resnet101 with batch size 32

                            sec/iter    ex/sec      sec/iter    ex/sec      sec/iter    ex/s
   1 GPUs --   no ddp:  p50:  0.162s     197/s  p75:  0.162s     197/s  p90:  0.162s     197
   1 GPUs --    1M/1G:  p50:  0.171s     187/s  p75:  0.171s     186/s  p90:  0.171s     186
   2 GPUs --    1M/2G:  p50:  0.176s     182/s  p75:  0.176s     181/s  p90:  0.176s     181
   4 GPUs --    1M/4G:  p50:  0.176s     182/s  p75:  0.176s     181/s  p90:  0.176s     181
   8 GPUs --    1M/8G:  p50:  0.179s     179/s  p75:  0.179s     178/s  p90:  0.180s     178
  16 GPUs --    2M/8G:  p50:  0.179s     178/s  p75:  0.180s     177/s  p90:  0.183s     174

Benchmark: resnext50_32x4d with batch size 32

                            sec/iter    ex/sec      sec/iter    ex/sec      sec/iter    ex/s
   1 GPUs --   no ddp:  p50:  0.145s     220/s  p75:  0.145s     220/s  p90:  0.145s     220
   1 GPUs --    1M/1G:  p50:  0.147s     217/s  p75:  0.147s     217/s  p90:  0.148s     216
   2 GPUs --    1M/2G:  p50:  0.153s     209/s  p75:  0.153s     209/s  p90:  0.153s     209
   4 GPUs --    1M/4G:  p50:  0.153s     208/s  p75:  0.153s     208/s  p90:  0.154s     208
   8 GPUs --    1M/8G:  p50:  0.157s     204/s  p75:  0.157s     204/s  p90:  0.157s     203
  16 GPUs --    2M/8G:  p50:  0.157s     203/s  p75:  0.157s     203/s  p90:  0.158s     203

Benchmark: resnext101_32x8d with batch size 32

                            sec/iter    ex/sec      sec/iter    ex/sec      sec/iter    ex/s
   1 GPUs --   no ddp:  p50:  0.415s      77/s  p75:  0.415s      77/s  p90:  0.416s      76
   1 GPUs --    1M/1G:  p50:  0.425s      75/s  p75:  0.426s      75/s  p90:  0.426s      75
   2 GPUs --    1M/2G:  p50:  0.438s      73/s  p75:  0.439s      72/s  p90:  0.439s      72
   4 GPUs --    1M/4G:  p50:  0.439s      72/s  p75:  0.439s      72/s  p90:  0.440s      72
   8 GPUs --    1M/8G:  p50:  0.447s      71/s  p75:  0.447s      71/s  p90:  0.448s      71
  16 GPUs --    2M/8G:  p50:  0.450s      71/s  p75:  0.451s      70/s  p90:  0.451s      70
```

How to diff

Run the benchmark with the `--json PATH_TO_REPORT_FILE` argument to produce the JSON file that the diff script can consume.

Then, run the diff script as follows:

```
$ python3 diff.py PATH_TO_BASELINE_FILE PATH_TO_TEST_FILE
                                 baseline                      test
                     --------------------      --------------------
bucket_size:                           25  vs                     1
cuda_version:                        10.0  vs                  10.0
distributed_backend:                 nccl  vs                  nccl
pytorch_version:          1.4.0a0+05140f0  vs       1.4.0a0+05140f0

Benchmark: resnet50 with batch size 32

                  sec/iter    ex/sec      diff        sec/iter    ex/sec      diff
   1 GPUs:  p75:    0.101s     317/s     -0.3%  p95:    0.101s     317/s     -0.4%
   2 GPUs:  p75:    0.104s     306/s     -1.0%  p95:    0.104s     306/s     -1.0%
   4 GPUs:  p75:    0.105s     305/s     -1.6%  p95:    0.105s     304/s     -1.8%
   8 GPUs:  p75:    0.107s     299/s     -2.6%  p95:    0.107s     298/s     -2.7%
  16 GPUs:  p75:    0.108s     294/s     -3.8%  p95:    0.122s     262/s    -16.4%

Benchmark: resnet101 with batch size 32

                  sec/iter    ex/sec      diff        sec/iter    ex/sec      diff
   1 GPUs:  p75:    0.172s     185/s     -1.2%  p95:    0.172s     185/s     -1.3%
   2 GPUs:  p75:    0.179s     178/s     -2.1%  p95:    0.179s     178/s     -2.0%
   4 GPUs:  p75:    0.180s     177/s     -2.6%  p95:    0.180s     177/s     -2.6%
   8 GPUs:  p75:    0.184s     173/s     -3.5%  p95:    0.184s     173/s     -3.5%
  16 GPUs:  p75:    0.187s     170/s     -0.1%  p95:    0.204s     157/s     -7.9%

Benchmark: resnext50_32x4d with batch size 32

                  sec/iter    ex/sec      diff        sec/iter    ex/sec      diff
   1 GPUs:  p75:    0.149s     214/s     -1.0%  p95:    0.149s     214/s     -0.9%
   2 GPUs:  p75:    0.156s     205/s     -1.5%  p95:    0.156s     205/s     -1.6%
   4 GPUs:  p75:    0.156s     204/s     -1.6%  p95:    0.157s     204/s     -1.8%
   8 GPUs:  p75:    0.159s     200/s     -1.5%  p95:    0.159s     200/s     -1.5%
  16 GPUs:  p75:    0.161s     198/s     -1.9%  p95:    0.162s     197/s     -2.3%

Benchmark: resnext101_32x8d with batch size 32

                  sec/iter    ex/sec      diff        sec/iter    ex/sec      diff
   1 GPUs:  p75:    0.427s      74/s     -0.8%  p95:    0.428s      74/s     -0.7%
   2 GPUs:  p75:    0.444s      72/s     -1.3%  p95:    0.445s      71/s     -0.7%
   4 GPUs:  p75:    0.444s      72/s     -1.1%  p95:    0.445s      71/s     -0.8%
   8 GPUs:  p75:    0.452s      70/s     -1.3%  p95:    0.452s      70/s     -1.3%
  16 GPUs:  p75:    0.455s      70/s     -0.7%  p95:    0.456s      70/s     -0.6%
```

This compares throughput between `bucket_cap_mb=25` (the default) and `bucket_cap_mb=1` on 8 DGX machines with V100 GPUs. It confims that even for a relatively small model on machines with a very fast interconnect (4x 100Gb InfiniBand per machine), it still pays off to batch allreduce calls.