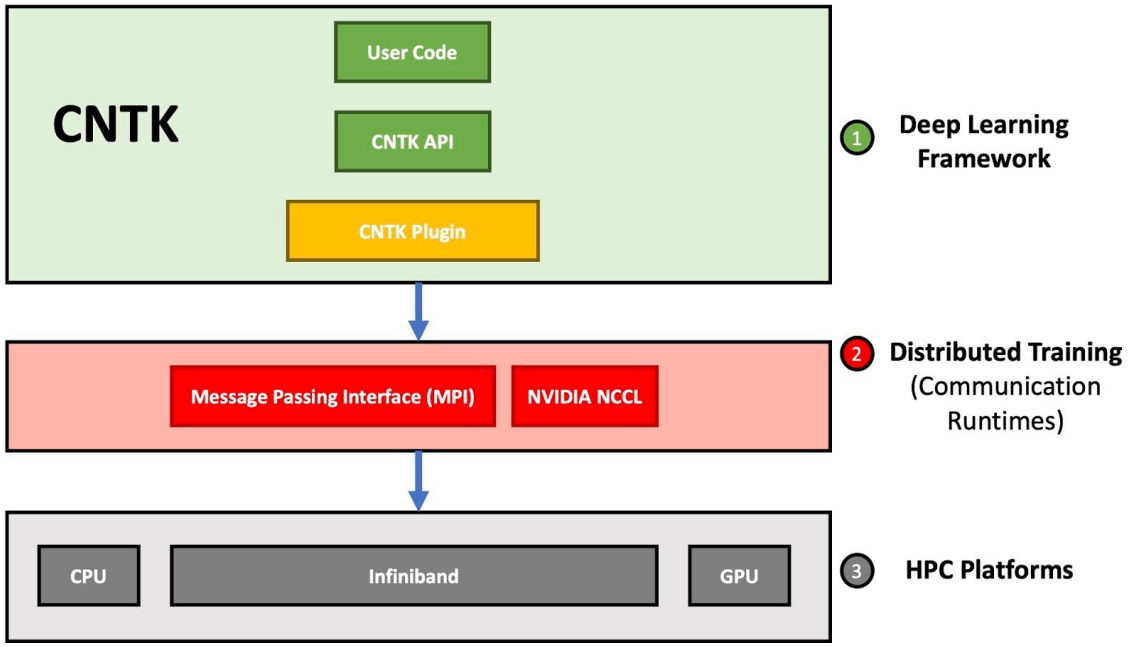
5194 HiDL Lab 2 Report - **CNTK**

Christopher Britt; Shuangsheng Lou; Anthony J. Cavallero; Kawthar Shafie Khorassani

1. **(20 points) Investigate one or two options for running distributed (multi-node) training for your assigned Deep Learning framework and explain your understanding by highlighting major components/technologies needed for it.**

   a. Write a paragraph and draw a block diagram to explain your understanding.
   b. If there are multiple choice, list them down and highlight which approach have you used. E.g. MPI or gRPC or a combination of both for TensorFlow



In order to run CNTK on multiple nodes (distributed training), we utilized the Message Passing Interface (MPI). MPI communication enables executing applications at scale. It is often used in parallel applications to enable communication amongst processes. In relation to deep learning frameworks, we scale-up and scale-out. In scaling-up, we run multiple process within one node across multiple machines (intra-node), and in scaling-out, we run processes across multiple nodes of a cluster (inter-node). The challenge in scaling is being able to combine the benefits of both scaling up and scaling out (e.g. using many GPUs distributed across a cluster). In this lab, we evaluate multi-node training for CNTK by running experiments on multiple GPU nodes (e.g. 1,2,4 and 8 nodes) to evaluate the impact on throughput/performance as we scale out to a larger number of nodes. We also evaluated CPU performance through utilizing all the CPU cores of a node and through scaling out to multiple nodes.

Parallelization in CNTK is implemented using MPI. Each of the three models used in this lab (AlexNet, ResNet, and VGG16) can be distributed through running with mpi and specifying a hostfile related to the number of processes being run. CNTK also utilizes the NVIDIA Collective Communications Library (NCCL). It provides primitives for collectives communication on multiple GPUs. CNTK provides a configure option to enable usage of NCCL. According to the developers site, CNTK support for NCCL is limited to data-parallel SGD with 32/64 gradient bits (https://docs.microsoft.com/en-us/cognitive-toolkit/setup-gpu-specific-packages-linux). However, the CNTK release by default is built with support for NCCL.

To elaborate further on this in the future for lab 3, we will evaluate the specific communication patterns or which MPI calls are being used (e.g. allreduce) in CNTK through profiling and looking deeper into the code.

2. **Choose one of the following three large-scale datasets.**
   a. ImageNet (ILSVRC 2012 - ~200 GB dataset) -- http://image-net.org

We used tiny-ImageNet here instead of ImageNet, which is big enough for this research study.

3. **Choose three different DNN models available for your chosen dataset.**
   a. AlexNet, ResNet, and MobileNet for ImageNet dataset

We utilized Alexnet, Resnet, and VGG16 for our performance testing. Due to limitations of the CNTK library the necessary building blocks to implement an efficient MobileNet are not present. As such we have elected with permission from the instructional team to utilize VGG16 as a replacement.

4. **(20 points) Run the experiments for each DNN using a single node (If you have these numbers from Lab #1, please re-use these to save SUs. Otherwise, you can re-run these experiments.)**
   a. Run the GPU version

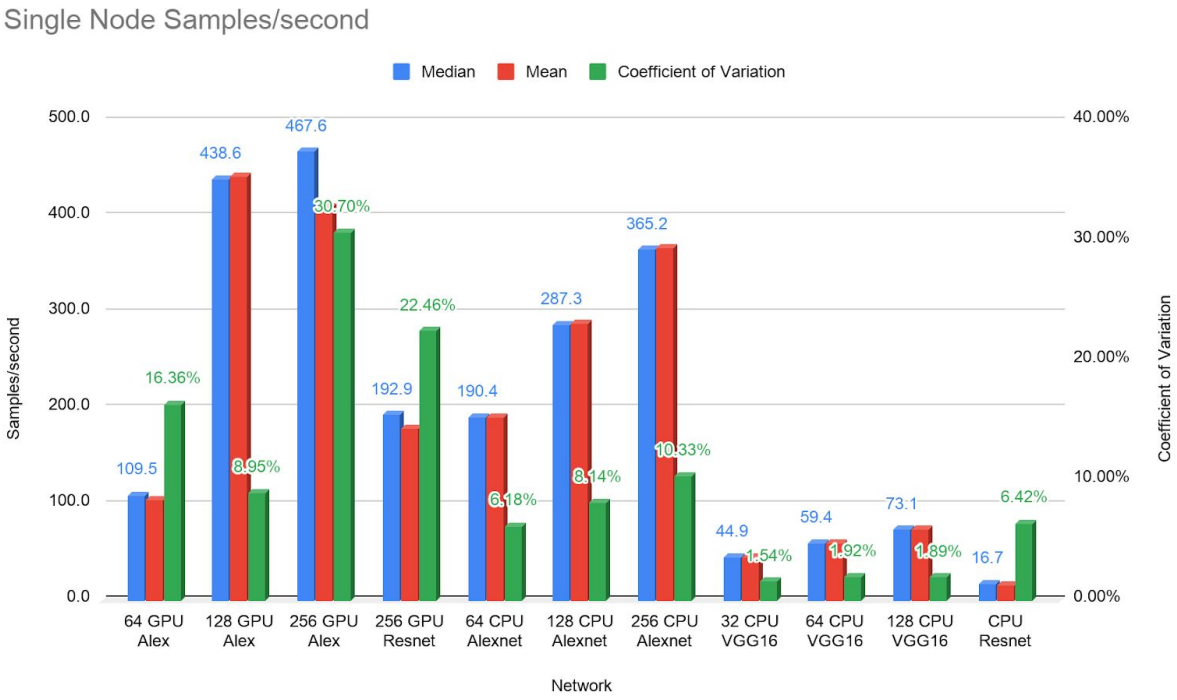
See GPU Appendix: [AlexNet: Appendix A-GPU, B-GPU, and C-GPU. ResNet: G-GPU. VGG16: K-GPU]

   b. Run the CPU version

See CPU Appendix: [AlexNet: Appendix A-CPU, B-CPU, and C-CPU. ResNet: G-CPU. VGG16: L-CPU, M-CPU, N-CPU and O-CPU]

   c. Report throughput (training time) in terms of samples/second or images/second for both CPU and GPU

For AlexNet on GPU we found the optimal batch size of 256 was able to process over 10 epochs an average of 467 samples/second, whereas on the CPU had a mean of 367 samples/second. By utilizing the GPU we were thus able to obtain a 27.2% performance increase in training Alexnet. While working on Resnet the divergence was even greater with the GPU performing an entire magnitude faster than the CPU.



Graph 1: Comparing GPU and CPU batch size performance variations across networks

d. Vary the batch size and find out the best batch size that gives you the highest throughput.

During our testing for Alexnet we decided to use 128 batch size on GPU's as it gave us the most stable best performance. We found this batch size provided a better balance between loss and throughput when compared to other batch sizes (e.g. 64 and 256 shown in GPU Appendix). As we moved away from our tested optimal batch size we noticed not only a drop in average performance, but also a significant increase in the coefficient of variation. The CNTK recommended Resnet implementation automatically computes the optimal batch size based on system and runtime parameters so we elected to use this system during our testing. For VGG we found the batch size of 128 produced the best performance in the network. On CPU, varying the batch size from 32 to 256 (e.g. 64, 128 and 256 for AlexNet, 32, 64, 128 and 256 for VGG), the best batch size chosen for AlexNet is 256 and the best batch size chosen for VGG-16 is 128, as shown both in graph1 and the table below.

Epoch #	64 GPU Alex	128 GPU Alex	256 GPU Alex	256 GPU Resnet	64 CPU Alexnet
1	61.4	368.8	120.1	65.1	174.4

2	90.7	406	257.5	193.6	175.9
3	106.3	426.4	373.4	188.7	208.8
4	113.6	440.7	453.4	184	202.4
5	119.9	464.5	462.9	193.6	189
6	116.5	436.5	472.2	195.7	204.4
7	108.4	454.6	474.2	193.1	189.8
8	112.6	425.2	486.5	194.9	195
9	110.5	502	501.5	191.5	191
10	106.4	492.9	490.5	192.7	180.7
Median	109.5	438.6	467.6	192.9	190.4
Mean	104.6	441.8	409.2	179.3	191.1
Standard Deviation	17.11	39.53	125.61	40.27	11.82
Coefficient of Variation	16.36%	8.95%	30.70%	22.46%	6.18%

Epoch #	128 CPU Alexnet	256 CPU Alexnet	32 CPU VGG16	64 CPU VGG16	128 CPU VGG16	32 CPU Resnet
1	320.8	311.0	45.7	60.5	72.7	14.4
2	284.3	354.63	44.9	59.8	74.7	15.9
3	254.9	412.5	44.1	58.4	73.3	16.5
4	258	403.4	45.7	59.1	75	16.9
5	271.7	358.7	44.4	57.8	74.1	15.3
6	286.9	418.5	44.8	59.2	71.7	16.8
7	324.1	371.7	45.2	60.7	72.8	14.3
8	304.2	345.8	44	59.6	73.8	16.8
9	287.6	378.5	44.4	57.6	70.7	16.8
10	296.1	313.1	45.9	60.7	71.9	16.8
Median	238.8	301.2	44.9	59.4	73.1	16.7
Mean	288.9	366.7	44.9	59.3	73.1	16.1
Standard Deviation	233.57	224.96	0.69	1.14	1.38	1.03
Coefficient of Variation	75.06%	65.13%	1.54%	1.92%	1.89%	6.42%

5. **(40 points) After you have the data for the best batch size for a single node, run the experiment with the best batch size for 1, 2, 4, and 8 nodes.**
    a. Run for CPU only

See CPU Appendix.[AlexNet: Appendix C-CPU, D-CPU, E-CPU and F-CPU. ResNet: Appendix G-CPU, H-CPU, I-CPU and J-CPU. VGG16: M-CPU, O-CPU, P-CPU and Q-GPU]

    b. Run for GPU only

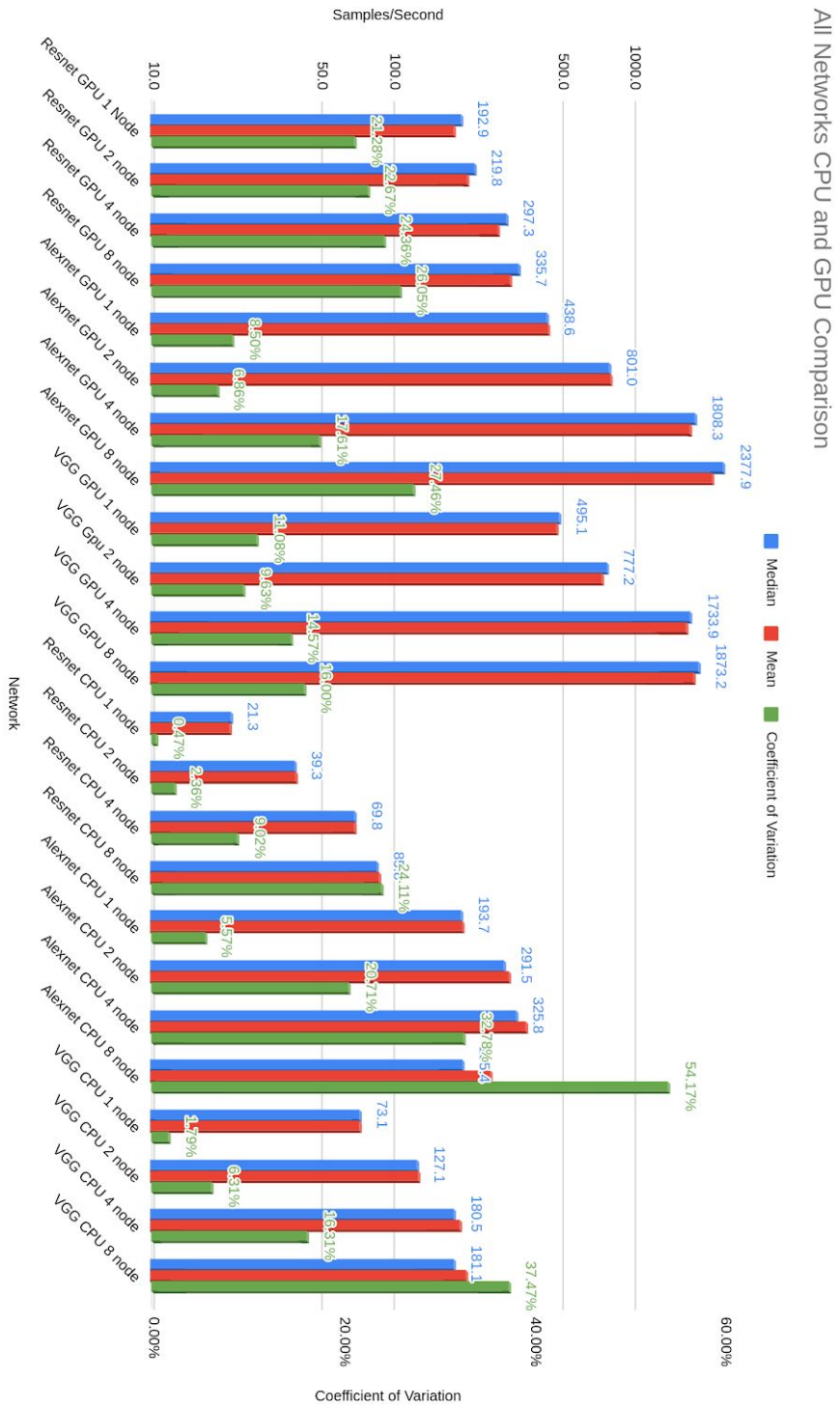
See GPU Appendix: [AlexNet: Appendix D-GPU, E-GPU, and F-GPU. ResNet: Appendix H-GPU, I-GPU, and J-GPU. VGG16: K-GPU, L-GPU, M-GPU, and N-GPU]

    c. Report the throughput for both versions

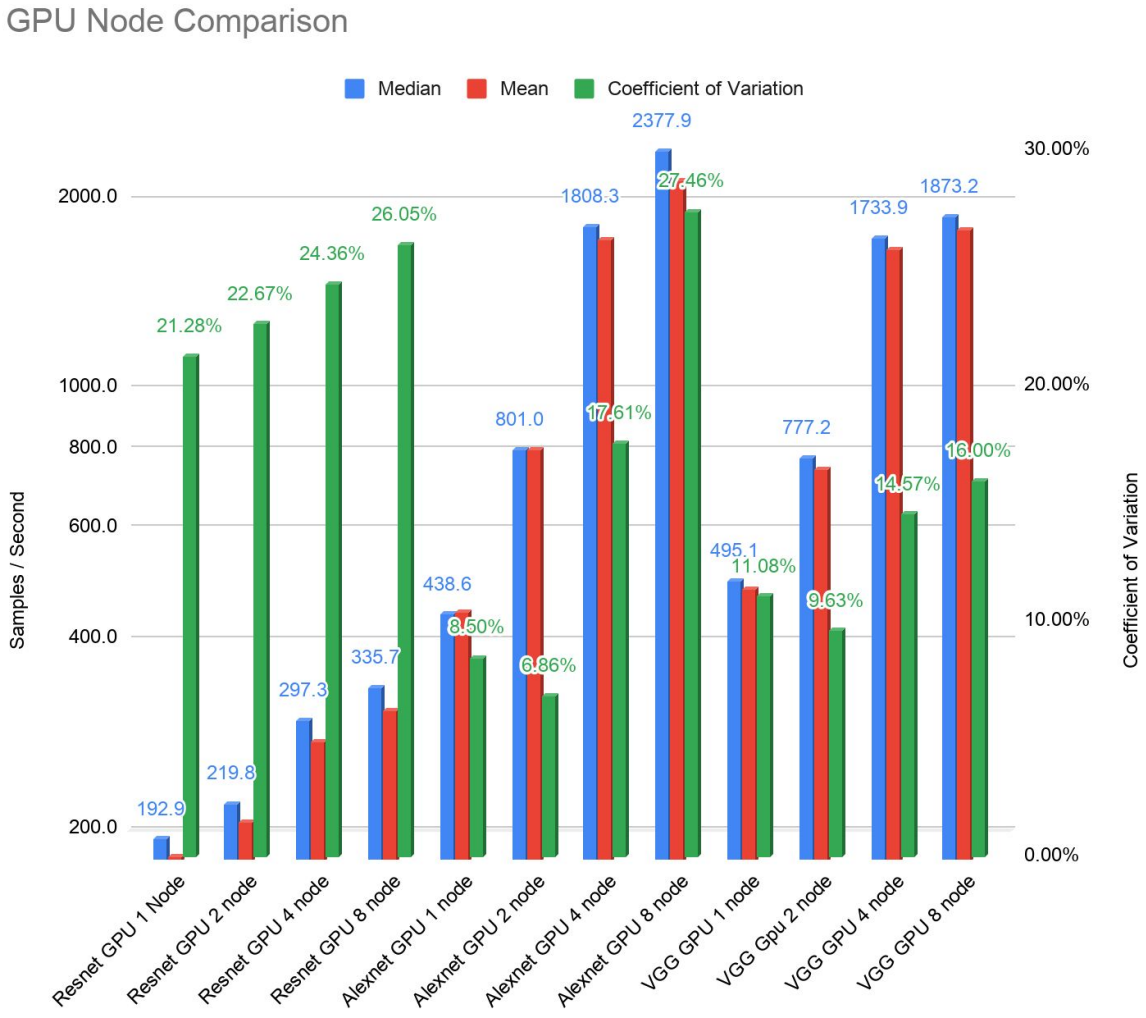
During our testing we found in almost all cases the performance increased as we scaled the number of nodes outwards, however the observed efficiency differed between networks. In addition, as can be seen in the following graphs we observed a strong positive correlation between node count and coefficient of variation; indicating there may be some instability and randomness encountered during the communication procedures. We also found the scalability of the GPU system far surpassed the scalability of the CPU systems when scaling to 8 or more nodes as communication overhead began reducing the overall performance of both alexnet and VGG.

    d. Report the scalability/speedup for multiple nodes by creating a graph that presents images/second on the y-axis and #nodes on x-axis.

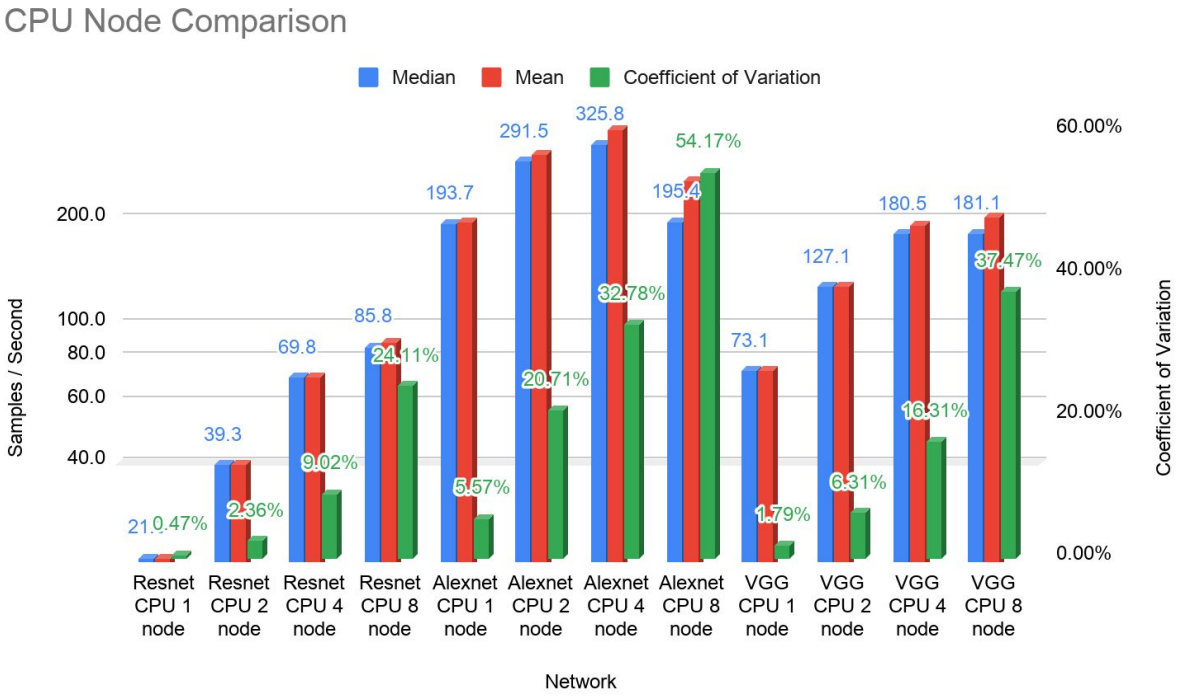
The following graphs demonstrate both CPU/GPU comparisons as well as network variations across different node counts.



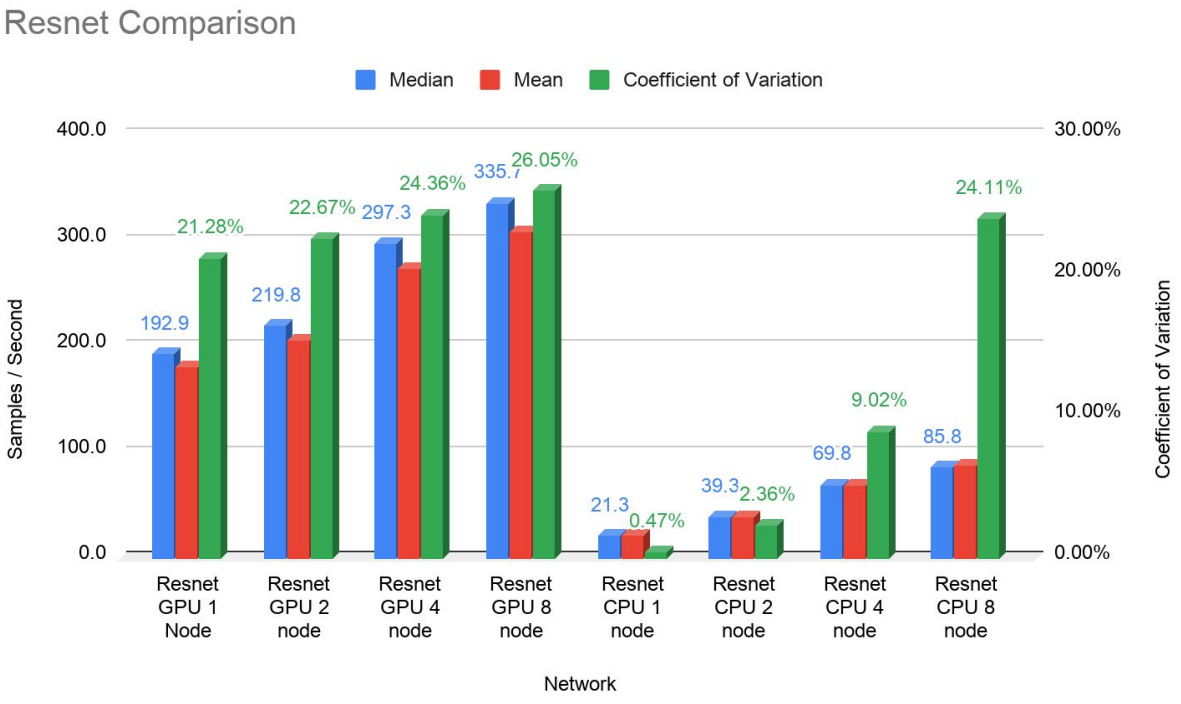
Graph 2: Performance and CV across GPU and CPU nodes



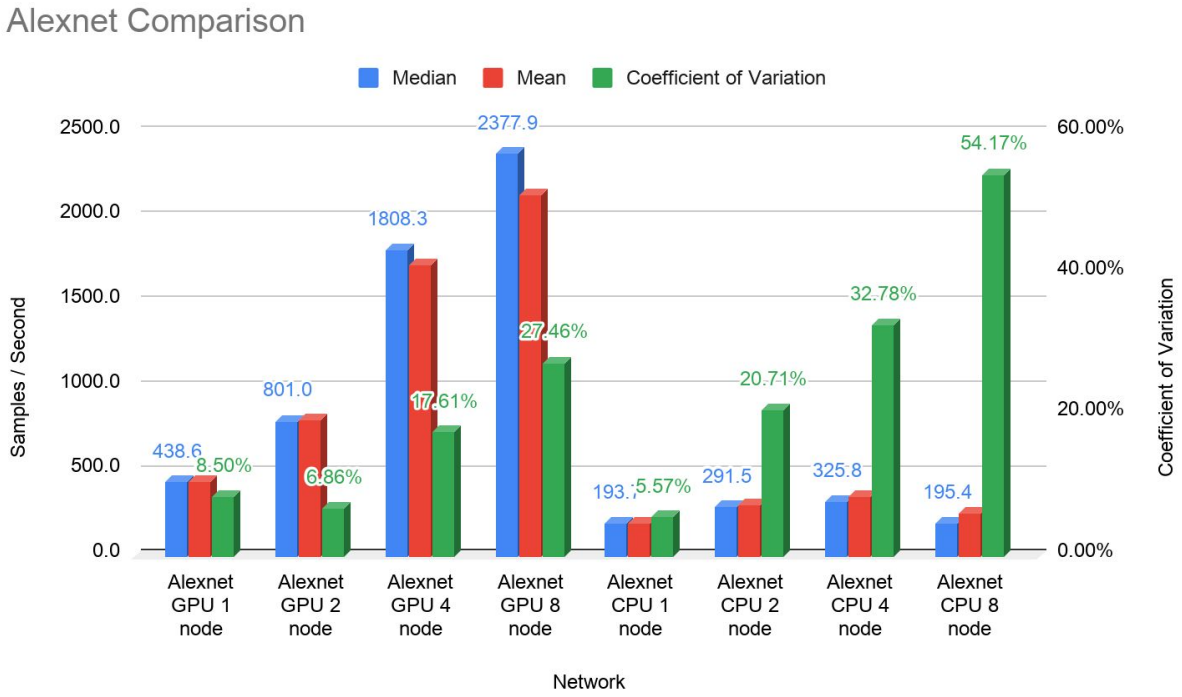
Graph 3: Comparing performance and CV across the GPU nodes



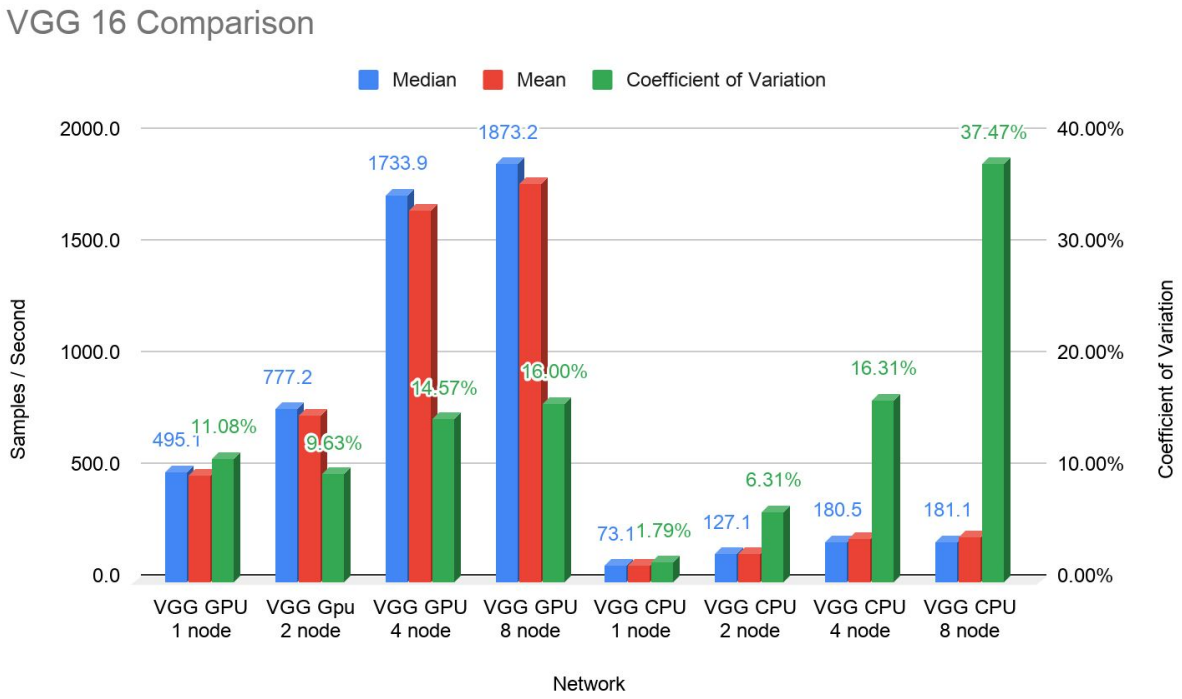
Graph 4: Comparing performance and CV across the CPU nodes



Graph 5: Comparing performance and CV of the ResNET network



Graph 6: Comparing performance and CV of the Alexnet network



Graph 7: Comparing performance and CV of the VGG 16 network

Multi node performance comparison table

Network	Median	Mean	Coefficient of Variation
Resnet GPU 1 Node	192.9	180.5	21.28%
Resnet GPU 2 node	219.8	205.7	22.67%
Resnet GPU 4 node	297.3	274.6	24.36%
Resnet GPU 8 node	335.7	307.8	26.05%
Alexnet GPU 1 node	438.6	441.5	8.50%
Alexnet GPU 2 node	801.0	801.3	6.86%
Alexnet GPU 4 node	1808.3	1723.2	17.61%
Alexnet GPU 8 node	2377.9	2134.9	27.46%
VGG GPU 1 node	495.1	480.0	11.08%
VGG Gpu 2 node	777.2	745.3	9.63%
VGG GPU 4 node	1733.9	1665.0	14.57%
VGG GPU 8 node	1873.2	1780.9	16.00%
Resnet CPU 1 node	21.3	21.2	0.47%
Resnet CPU 2 node	39.3	39.6	2.36%
Resnet CPU 4 node	69.8	69.9	9.02%
Resnet CPU 8 node	85.8	88.2	24.11%
Alexnet CPU 1 node	193.7	366.7	5.57%
Alexnet CPU 2 node	291.5	515.3	20.71%
Alexnet CPU 4 node	325.8	987.6	32.78%
Alexnet CPU 8 node	195.4	1628.9	54.17%
VGG CPU 1 node	73.1	73.1	1.79%
VGG CPU 2 node	127.1	127.7	6.31%

VGG CPU 4 node	180.5	191.8	16.31%
VGG CPU 8 node	181.1	201.8	37.47%

6. **(20 points) What can you conclude from this study?**
   a. Write a few paragraphs to explain your results and the insights in-depth.

I. Impact of Distributed Training

In lab 1 we had evaluated the performance of the selected models using our deep learning framework (CNTK) on a single node. We did additional experiments in this lab to analyze the impact of distributed training on the throughput returned by these tests. We evaluated the same experiments by varying batch size to find a good balance between our loss and the throughput. We then used the best configuration we can find on each of the models and on CPU/GPU to evaluate 1,2,4 and 8 nodes. We used MPI as the communication runtime to distribute the processes over multiple nodes. We found improved performance in most cases when scaling-outwards (inter-node). This trend can be observed in question 5 where 8-node GPU performance scales much better than 2-node GPU performance as evident in the throughput.

II. Impact of batch size

During this lab we investigated the impact of the selected batch size on our chosen models. The first thing of note to report is that the optimal batch size isn't static across the different models that we had tested. VGG, AlexNet, and ResNet all had a different optimal batch size, but what is unique to all three is how a given size is found to be optimal. We determined that we can claim a batch size is optimal based on two factors, maximization of throughput/images processed per second and minimization of the coefficient of variation. Since we are not training long enough to develop a metric in model accuracy, it will be omitted but in a typical discussion it would be included as a third factor. The general trend that we noticed in the AlexNet gpu portion of the question 4 graph was that for the smallest batch size there wasn't a very large throughput relative to the variation of the total throughputs. However, using a large batch size despite providing a high throughput also introduced a large variance into the total throughputs we observed. In this case 128 was an optimal batch size where we had a high throughput for the model while also being stable compared to the largest size. This trend can be noticed in the CPU run as well. It should also be noted that the trend can be noticed in the other models with their respective optimal batch sizes if we were to introduce accuracy in this conversation as a general assumption. When we use small batch sizes, we update our model more frequently, allowing for a more accurate model while sacrificing training time and throughput of the model. On the other hand, if we choose a large batch size, we update our model less frequently, which sacrifices the accuracy of the model in order to have a higher throughput and shorter training time. If we were to add in this third metric, we would need to find the batch size that provides the

highest throughput, the smallest variation of throughput, and does not cause a large impact on accuracy of the model as a whole.

III. Overall performance evaluation

During the experiment we observed increased network performance from scaling 1 to 8 nodes for the GPU runs, along with CPU Resnet and CPU VGG as can be seen in graphs 2, 3, and 4. This indicates the computational workload remained substantial enough to outweigh the added communication overhead as we scaled the number of nodes upwards. However, for Alexnet CPU we observed a performance drop when moving from 4 to 8 nodes, indicating the communication overhead had surpassed the computational requirements of the model. If we desired to further increase the scalability of Alexnet on multiple CPU nodes we would have to sacrifice further accuracy to further raise the batch size, thus increasing the computational workload performed between each synchronization.

CPU Runs Appendix:

Appendix A - Running Alexnet on **CPU** with 10k epoch size, 64 batch size per core,1 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python AlexNet_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 1 --minibatch_size 1280

Finished Epoch[1 of 1]: [Training] loss = 5.303679 * 10000, metric = 99.38% * 10000 57.345s (174.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303679 * 10000, metric = 99.38% * 10000 56.863s (175.9 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303679 * 10000, metric = 99.38% * 10000 47.882s (208.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303679 * 10000, metric = 99.38% * 10000 49.396s (202.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303679 * 10000, metric = 99.38% * 10000 52.897s (189.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303679 * 10000, metric = 99.38% * 10000 48.921s (204.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303679 * 10000, metric = 99.38% * 10000 52.680s (189.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303679 * 10000, metric = 99.38% * 10000 51.290s (195.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303679 * 10000, metric = 99.38% * 10000 52.351s (191.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303679 * 10000, metric = 99.38% * 10000 55.352s (180.7 samples/s);

...

Appendix B - Running Alexnet on **CPU** with 10k epoch size, 128 batch size per core,1 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python AlexNet_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 1 --minibatch_size 2560

Finished Epoch[1 of 1]: [Training] loss = 5.304305 * 10000, metric = 99.41% * 10000 31.174s (320.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304305 * 10000, metric = 99.41% * 10000 35.180s (284.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304305 * 10000, metric = 99.41% * 10000 39.228s (254.9 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304305 * 10000, metric = 99.41% * 10000 38.767s (258.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304305 * 10000, metric = 99.41% * 10000 36.810s (271.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304305 * 10000, metric = 99.41% * 10000 34.857s (286.9 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304305 * 10000, metric = 99.41% * 10000 30.858s (324.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304305 * 10000, metric = 99.41% * 10000 32.871s (304.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304305 * 10000, metric = 99.41% * 10000 34.774s (287.6 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304305 * 10000, metric = 99.41% * 10000 33.777s (296.1 samples/s);

...

Appendix C - Running Alexnet on **CPU** with 10k epoch size, 256 batch size per core, 1 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python AlexNet_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 1 --minibatch_size 5120

Finished Epoch[1 of 1]: [Training] loss = 5.304485 * 10000, metric = 99.42% * 10000 32.154s (311.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304485 * 10000, metric = 99.42% * 10000 28.200s (354.6 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304485 * 10000, metric = 99.42% * 10000 24.242s (412.5 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304485 * 10000, metric = 99.42% * 10000 24.788s (403.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304485 * 10000, metric = 99.42% * 10000 27.880s (358.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304485 * 10000, metric = 99.42% * 10000 23.893s (418.5 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304485 * 10000, metric = 99.42% * 10000 26.905s (371.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304485 * 10000, metric = 99.42% * 10000 28.918s (345.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304485 * 10000, metric = 99.42% * 10000 26.418s (378.5 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304485 * 10000, metric = 99.42% * 10000 31.943s (313.1 samples/s);

...

Appendix D - Running Alexnet on **CPU** with 10k epoch size, 256 batch size per core, 2 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python AlexNet_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 1 --minibatch_size 10240

Finished Epoch[1 of 1]: [Training] loss = 5.305711 * 10000, metric = 99.51% * 10000 12.835s (779.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.305711 * 10000, metric = 99.51% * 10000 12.360s (809.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.305711 * 10000, metric = 99.51% * 10000 20.423s (489.6 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.305711 * 10000, metric = 99.51% * 10000 10.997s (909.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.305711 * 10000, metric = 99.51% * 10000 29.057s (344.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.305711 * 10000, metric = 99.51% * 10000 28.558s (350.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.305711 * 10000, metric = 99.51% * 10000 24.229s (412.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.305711 * 10000, metric = 99.51% * 10000 21.237s (470.9 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.305711 * 10000, metric = 99.51% * 10000 16.287s (614.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.305711 * 10000, metric = 99.51% * 10000 19.302s (518.1 samples/s);

...

Appendix E - Running Alexnet on **CPU** with 40k epoch size, 256 batch size per core, 4 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python AlexNet_ImageNet_Distributed.py --epoch_size 40000 --num_epochs 1 --minibatch_size 20480

Finished Epoch[1 of 1]: [Training] loss = 5.304025 * 40000, metric = 99.42% * 40000 29.665s (1348.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304025 * 40000, metric = 99.42% * 40000 42.180s (948.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304025 * 40000, metric = 99.42% * 40000 45.707s (875.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304025 * 40000, metric = 99.42% * 40000 53.754s (744.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304025 * 40000, metric = 99.42% * 40000 46.266s (864.6 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304025 * 40000, metric = 99.42% * 40000 58.268s (686.5 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304025 * 40000, metric = 99.42% * 40000 52.285s (765.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304025 * 40000, metric = 99.42% * 40000 62.294s (642.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304025 * 40000, metric = 99.42% * 40000 36.302s (1101.9 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304025 * 40000, metric = 99.42% * 40000 41.351s (967.3 samples/s);

...

Appendix F - Running Alexnet on **CPU** with 80k epoch size, 256 batch size per core, 8 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python AlexNet_ImageNet_Distributed.py --epoch_size 80000 --num_epochs 1 --minibatch_size 40960

Finished Epoch[1 of 1]: [Training] loss = 5.303289 * 80000, metric = 99.53% * 80000 54.498s (1467.9 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303289 * 80000, metric = 99.53% * 80000 44.998s (1777.9 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303289 * 80000, metric = 99.53% * 80000 51.998s (1538.5 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303289 * 80000, metric = 99.53% * 80000 88.000s (909.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303289 * 80000, metric = 99.53% * 80000 102.506s (780.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303289 * 80000, metric = 99.53% * 80000 64.510s (1240.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303289 * 80000, metric = 99.53% * 80000 96.513s (828.9 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303289 * 80000, metric = 99.53% * 80000 53.017s (1508.9 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303289 * 80000, metric = 99.53% * 80000 42.021s (1903.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.303289 * 80000, metric = 99.53% * 80000 24.522s (3262.4 samples/s);

...

Appendix G - Running Resnet on **CPU** with 10k epoch size, 32 batch size per core, 1 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python TrainResNet_ImageNet_Distributed.py --network resnet34 --epochs 1 --epoch_size 10000 --scale_up True

---

Finished Epoch[1 of 1]: [Training] loss = 5.846385 * 10000, metric = 99.37% * 10000 471.615s (21.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.846385 * 10000, metric = 99.37% * 10000 473.115s (21.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.846385 * 10000, metric = 99.37% * 10000 469.125s (21.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.846385 * 10000, metric = 99.37% * 10000 473.626s (21.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.846385 * 10000, metric = 99.37% * 10000 470.126s (21.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.846385 * 10000, metric = 99.37% * 10000 469.626s (21.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.846385 * 10000, metric = 99.37% * 10000 468.626s (21.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.846385 * 10000, metric = 99.37% * 10000 474.627s (21.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.846385 * 10000, metric = 99.37% * 10000 466.626s (21.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.846385 * 10000, metric = 99.37% * 10000 472.127s (21.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.846385 * 10000, metric = 99.37% * 10000 468.127s (21.4 samples/s);

…

Appendix H - Running Resnet on **CPU** with 10k epoch size, 32 batch size per core, 2 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python TrainResNet_ImageNet_Distributed.py --network resnet34 --epochs 1 --epoch_size 10000 --scale_up True

---

Finished Epoch[1 of 1]: [Training] loss = 6.328596 * 10000, metric = 99.48% * 10000 259.197s (38.6 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 6.328596 * 10000, metric = 99.48% * 10000 242.698s (41.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 6.328596 * 10000, metric = 99.48% * 10000 255.198s (39.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 6.328596 * 10000, metric = 99.48% * 10000 254.198s (39.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 6.328596 * 10000, metric = 99.48% * 10000 258.198s (38.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 6.328596 * 10000, metric = 99.48% * 10000 256.198s (39.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 6.328596 * 10000, metric = 99.48% * 10000 249.198s (40.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 6.328596 * 10000, metric = 99.48% * 10000 257.198s (38.9 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 6.328596 * 10000, metric = 99.48% * 10000 242.199s (41.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 6.328596 * 10000, metric = 99.48% * 10000 251.699s (39.7 samples/s);

…

Appendix I - Running Resnet on **CPU** with 10k epoch size, 32 batch size per core, 4 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python TrainResNet_ImageNet_Distributed.py --network resnet34 --epochs 1 --epoch_size 10000 --scale_up True

---

Finished Epoch[1 of 1]: [Training] loss = 7.848047 * 10000, metric = 99.43% * 10000 131.239s (76.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 7.848047 * 10000, metric = 99.43% * 10000 157.746s (63.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 7.848047 * 10000, metric = 99.43% * 10000 157.248s (63.6 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 7.848047 * 10000, metric = 99.43% * 10000 137.749s (72.6 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 7.848047 * 10000, metric = 99.43% * 10000 127.249s (78.6 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 7.848047 * 10000, metric = 99.43% * 10000 130.249s (76.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 7.848047 * 10000, metric = 99.43% * 10000 160.750s (62.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 7.848047 * 10000, metric = 99.43% * 10000 158.750s (63.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 7.848047 * 10000, metric = 99.43% * 10000 149.250s (67.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 7.848047 * 10000, metric = 99.43% * 10000 132.750s (75.3 samples/s);

. . .

Appendix J - Running Resnet on **CPU** with 10k epoch size, 32 batch size per core, 8 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python TrainResNet_ImageNet_Distributed.py --network resnet34 --epochs 1 --epoch_size 10000 --scale_up True

---

Finished Epoch[1 of 1]: [Training] loss = 9.223289 * 10000, metric = 99.39% * 10000 146.783s (68.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 9.223289 * 10000, metric = 99.39% * 10000 138.783s (72.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 9.223289 * 10000, metric = 99.39% * 10000 141.783s (70.5 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 9.223289 * 10000, metric = 99.39% * 10000 140.283s (71.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 9.223289 * 10000, metric = 99.39% * 10000 114.783s (87.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 9.223289 * 10000, metric = 99.39% * 10000 92.783s (107.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 9.223289 * 10000, metric = 99.39% * 10000 70.783s (141.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 9.223289 * 10000, metric = 99.39% * 10000 106.283s (94.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 9.223289 * 10000, metric = 99.39% * 10000 118.283s (84.5 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 9.223289 * 10000, metric = 99.39% * 10000 114.283s (87.5 samples/s);

. . .

Appendix K - Running VGG16 on **CPU** with 10k epoch size, 32 batch size per core, 1 node

(local2) -bash-4.2$ time mpiexec --npernode 20 python VGG16_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 1 --minibatch_size 640

Finished Epoch[1 of 1]: [Training] loss = 5.301791 * 10000, metric = 99.54% * 10000 219.002s (45.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.301791 * 10000, metric = 99.54% * 10000 222.565s (44.9 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.301791 * 10000, metric = 99.54% * 10000 226.584s (44.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.301791 * 10000, metric = 99.54% * 10000 218.609s (45.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.301791 * 10000, metric = 99.54% * 10000 225.439s (44.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.301791 * 10000, metric = 99.54% * 10000 223.443s (44.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.301791 * 10000, metric = 99.54% * 10000 221.475s (45.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.301791 * 10000, metric = 99.54% * 10000 227.476s (44.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.301791 * 10000, metric = 99.54% * 10000 225.023s (44.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.301791 * 10000, metric = 99.54% * 10000 218.030s (45.9 samples/s);

...

Appendix L - Running VGG16 on **CPU** with 10k epoch size, 64 batch size per core, 1 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python VGG16_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 1 --minibatch_size 1280

Finished Epoch[1 of 1]: [Training] loss = 5.304491 * 10000, metric = 99.58% * 10000 165.252s (60.5 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304491 * 10000, metric = 99.58% * 10000 167.275s (59.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304491 * 10000, metric = 99.58% * 10000 171.311s (58.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304491 * 10000, metric = 99.58% * 10000 169.329s (59.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304491 * 10000, metric = 99.58% * 10000 172.867s (57.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304491 * 10000, metric = 99.58% * 10000 168.875s (59.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304491 * 10000, metric = 99.58% * 10000 164.877s (60.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304491 * 10000, metric = 99.58% * 10000 167.877s (59.6 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304491 * 10000, metric = 99.58% * 10000 173.693s (57.6 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.304491 * 10000, metric = 99.58% * 10000 164.795s (60.7 samples/s);

...

Appendix M - Running VGG16 on **CPU** with 10k epoch size, 128 batch size per core, 1 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python VGG16_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 1 --minibatch_size 2560

Finished Epoch[1 of 1]: [Training] loss = 5.308250 * 10000, metric = 99.51% * 10000 137.579s (72.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.308250 * 10000, metric = 99.51% * 10000 133.863s (74.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.308250 * 10000, metric = 99.51% * 10000 136.365s (73.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.308250 * 10000, metric = 99.51% * 10000 133.374s (75.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.308250 * 10000, metric = 99.51% * 10000 134.924s (74.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.308250 * 10000, metric = 99.51% * 10000 139.432s (71.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.308250 * 10000, metric = 99.51% * 10000 137.446s (72.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.308250 * 10000, metric = 99.51% * 10000 135.468s (73.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.308250 * 10000, metric = 99.51% * 10000 141.477s (70.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.308250 * 10000, metric = 99.51% * 10000 139.017s (71.9 samples/s);
…

real 3m44.169s
user 50m55.557s
sys 36m21.751s

Appendix N - Running VGG16 on **CPU** with 10k epoch size, 256 batch size per core, 1 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python VGG16_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 1 --minibatch_size 5120

---

real time is more than 7m23.011s

Appendix O - Running VGG16 on **CPU** with 10k epoch size, 128 batch size per core, 2 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python VGG16_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 1 --minibatch_size 5120

Finished Epoch[1 of 1]: [Training] loss = 5.310803 * 10000, metric = 99.45% * 10000 72.187s (138.5 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310803 * 10000, metric = 99.45% * 10000 71.826s (139.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310803 * 10000, metric = 99.45% * 10000 77.412s (129.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310803 * 10000, metric = 99.45% * 10000 75.934s (131.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310803 * 10000, metric = 99.45% * 10000 80.086s (124.9 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310803 * 10000, metric = 99.45% * 10000 80.606s (124.1 samples/s);

Finished Epoch[1 of 1]: [Training] loss = 5.310803 * 10000, metric = 99.45% * 10000 84.141s (118.8 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310803 * 10000, metric = 99.45% * 10000 81.647s (122.5 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310803 * 10000, metric = 99.45% * 10000 88.160s (113.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310803 * 10000, metric = 99.45% * 10000 74.190s (134.8 samples/s);

...

Appendix P - Running VGG16 on **CPU** with 10k epoch size, 128 batch size per core, 4 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python VGG16_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 1 --minibatch_size 10240

Finished Epoch[1 of 1]: [Training] loss = 5.307736 * 10000, metric = 99.43% * 10000 59.622s (167.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.307736 * 10000, metric = 99.43% * 10000 41.197s (242.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.307736 * 10000, metric = 99.43% * 10000 56.728s (176.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.307736 * 10000, metric = 99.43% * 10000 69.346s (144.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.307736 * 10000, metric = 99.43% * 10000 48.851s (204.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.307736 * 10000, metric = 99.43% * 10000 55.862s (179.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.307736 * 10000, metric = 99.43% * 10000 56.404s (177.3 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.307736 * 10000, metric = 99.43% * 10000 54.954s (182.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.307736 * 10000, metric = 99.43% * 10000 40.454s (247.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.307736 * 10000, metric = 99.43% * 10000 47.962s (208.5 samples/s);

...

Appendix Q - Running VGG16 on **CPU** with 20k epoch size, 128 batch size per core, 8 node

(local2) -bash-4.2$ time mpiexec --hostfile ./hfile python VGG16_ImageNet_Distributed.py --epoch_size 20000 --num_epochs 1 --minibatch_size 20480

Finished Epoch[1 of 1]: [Training] loss = 5.310652 * 20000, metric = 99.47% * 20000 95.190s (210.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310652 * 20000, metric = 99.47% * 20000 41.314s (484.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310652 * 20000, metric = 99.47% * 20000 90.817s (220.2 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310652 * 20000, metric = 99.47% * 20000 89.824s (222.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310652 * 20000, metric = 99.47% * 20000 94.324s (212.0 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310652 * 20000, metric = 99.47% * 20000 48.832s (409.6 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310652 * 20000, metric = 99.47% * 20000 93.838s (213.1 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310652 * 20000, metric = 99.47% * 20000 57.849s (345.7 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310652 * 20000, metric = 99.47% * 20000 48.377s (413.4 samples/s);
Finished Epoch[1 of 1]: [Training] loss = 5.310652 * 20000, metric = 99.47% * 20000 46.389s (431.1 samples/s);

...

GPU Runs Appendix:

Appendix A-GPU - Running Alexnet on **GPU** with 10k epoch size, 64 batch size, single node

(local-cntk-gpu) [kshafie@o0722 HiDL]$ python AlexNet_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 10 --minibatch_size 64

Selected GPU[0] Tesla P100-PCIE-16GB as the process wide default device.

Training 25171464 parameters in 16 parameter tensors.
Finished Epoch[1 of 10]: [Training] loss = 5.307815 * 10048, metric = 99.35% * 10048 163.654s (61.4 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.209617 * 9984, metric = 99.05% * 9984 110.119s (90.7 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.131053 * 9984, metric = 98.86% * 9984 93.891s (106.3 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.092217 * 9984, metric = 98.44% * 9984 87.926s (113.6 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.056322 * 10048, metric = 98.40% * 10048 83.813s (119.9 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.017490 * 9984, metric = 98.02% * 9984 85.670s (116.5 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 4.966154 * 9984, metric = 97.74% * 9984 92.134s (108.4 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.936625 * 9984, metric = 97.36% * 9984 88.680s (112.6 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.888680 * 10048, metric = 97.33% * 10048 90.905s (110.5 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.853004 * 9952, metric = 96.98% * 9952 93.571s (106.4 samples/s);
Finished Evaluation [1]: Minibatch[1-157]: metric = 96.99% * 10000;

Appendix B-GPU - Running Alexnet on **GPU** with 10k epoch size, 128 batch size, single node

(local-cntk-gpu) [kshafie@o0722 HiDL]$ python AlexNet_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 10 --minibatch_size 128

Selected GPU[0] Tesla P100-PCIE-16GB as the process wide default device.
Training 25171464 parameters in 16 parameter tensors.

Finished Epoch[1 of 10]: [Training] loss = 5.304100 * 10112, metric = 99.38% * 10112 27.417s (368.8 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.286130 * 9984, metric = 99.21% * 9984 24.594s (406.0 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.181125 * 9984, metric = 98.86% * 9984 23.415s (426.4 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.141124 * 9984, metric = 98.72% * 9984 22.657s (440.7 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.118026 * 9984, metric = 98.84% * 9984 21.493s (464.5 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.083585 * 9984, metric = 98.42% * 9984 22.875s (436.5 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.031957 * 9984, metric = 97.99% * 9984 21.961s (454.6 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.999450 * 9984, metric = 97.72% * 9984 23.480s (425.2 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.959669 * 10112, metric = 97.28% * 10112 20.142s (502.0 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.914295 * 9888, metric = 97.31% * 9888 20.059s (492.9 samples/s);
Finished Evaluation [1]: Minibatch[1-79]: metric = 97.14% * 10000;

Appendix C-GPU - Running Alexnet on **GPU** with 10k epoch size, 256 batch size, single node

(local-cntk-gpu) [kshafie@o0719 HiDL]$ python AlexNet_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 10

Selected GPU[0] Tesla P100-PCIE-16GB as the process wide default device.
Training 25171464 parameters in 16 parameter tensors.

Finished Epoch[1 of 10]: [Training] loss = 5.301822 * 10240, metric = 99.35% * 10240 85.245s (120.1 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.298998 * 9984, metric = 99.40% * 9984 38.774s (257.5 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.292076 * 9984, metric = 99.13% * 9984 26.740s (373.4 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.220207 * 9984, metric = 98.91% * 9984 22.021s (453.4 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.168812 * 9984, metric = 98.97% * 9984 21.569s (462.9 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.134045 * 9984, metric = 98.99% * 9984 21.144s (472.2 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.102593 * 9984, metric = 98.64% * 9984 21.055s (474.2 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.072197 * 9984, metric = 98.47% * 9984 20.521s (486.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.047056 * 9984, metric = 98.16% * 9984 19.907s (501.5 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.020217 * 9888, metric = 98.22% * 9888 20.158s (490.5 samples/s);

Finished Evaluation [1]: Minibatch[1-40]: metric = 97.92% * 10000;

Appendix D-GPU - Running Alexnet on **2 GPU Nodes**

(local-cntk-gpu) [kshafie@o0784 HiDL]$ mpiexec -n 2 --hostfile ./hfile python AlexNet_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 10 --minibatch_size 128
Finished Epoch[1 of 10]: [Training] loss = 5.304847 * 10112, metric = 99.31% * 10112 15.452s (654.4 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.304847 * 10112, metric = 99.31% * 10112 15.956s (633.7 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.295299 * 9984, metric = 99.40% * 9984 11.746s (850.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.295299 * 9984, metric = 99.40% * 9984 11.743s (850.2 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.189569 * 9984, metric = 98.94% * 9984 12.644s (789.6 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.189569 * 9984, metric = 98.94% * 9984 12.650s (789.2 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.137701 * 9984, metric = 98.67% * 9984 12.465s (801.0 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.137701 * 9984, metric = 98.67% * 9984 12.463s (801.1 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.094898 * 9984, metric = 98.68% * 9984 11.789s (846.9 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.094898 * 9984, metric = 98.68% * 9984 11.791s (846.7 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.045555 * 9984, metric = 98.25% * 9984 12.734s (784.0 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.045555 * 9984, metric = 98.25% * 9984 12.737s (783.9 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.008335 * 9984, metric = 97.86% * 9984 12.464s (801.0 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.008335 * 9984, metric = 97.86% * 9984 12.471s (800.6 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.985050 * 9984, metric = 97.54% * 9984 11.972s (833.9 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.985050 * 9984, metric = 97.54% * 9984 11.978s (833.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.941863 * 10112, metric = 97.51% * 10112 11.854s (853.0 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.941863 * 10112, metric = 97.51% * 10112 11.835s (854.4 samples/s);

Finished Epoch[10 of 10]: [Training] loss = 4.896113 * 9888, metric = 97.21% * 9888 12.360s (800.0 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.896113 * 9888, metric = 97.21% * 9888 12.365s (799.7 samples/s);
Finished Evaluation [1]: Minibatch[1-79]: metric = 96.95% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 96.95% * 10000;

Appendix E-GPU - Running Aexnet on **4 GPU Nodes**

(local-cntk-gpu) [kshafie@o0784 HiDL]$ mpiexec -n 4 --hostfile ./hfile python AlexNet_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 10 --minibatch_size 128
Finished Epoch[1 of 10]: [Training] loss = 5.303537 * 10112, metric = 99.39% * 10112 13.664s (740.0 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.303537 * 10112, metric = 99.39% * 10112 13.164s (768.2 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.303537 * 10112, metric = 99.39% * 10112 12.664s (798.5 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.303537 * 10112, metric = 99.39% * 10112 12.164s (831.3 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.290871 * 9984, metric = 99.34% * 9984 5.370s (1859.2 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.290871 * 9984, metric = 99.34% * 9984 5.370s (1859.2 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.290871 * 9984, metric = 99.34% * 9984 5.370s (1859.2 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.290871 * 9984, metric = 99.34% * 9984 5.370s (1859.2 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.181354 * 9984, metric = 98.86% * 9984 5.308s (1880.9 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.181354 * 9984, metric = 98.86% * 9984 5.308s (1880.9 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.181354 * 9984, metric = 98.86% * 9984 5.308s (1880.9 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.181354 * 9984, metric = 98.86% * 9984 5.308s (1880.9 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.140371 * 9984, metric = 98.63% * 9984 5.496s (1816.6 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.140371 * 9984, metric = 98.63% * 9984 5.496s (1816.6 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.140371 * 9984, metric = 98.63% * 9984 5.496s (1816.6 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.140371 * 9984, metric = 98.63% * 9984 5.496s (1816.6 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.104394 * 9984, metric = 98.65% * 9984 5.547s (1799.9 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.104394 * 9984, metric = 98.65% * 9984 5.547s (1799.9 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.104394 * 9984, metric = 98.65% * 9984 5.547s (1799.9 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.104394 * 9984, metric = 98.65% * 9984 5.547s (1799.9 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.062681 * 9984, metric = 98.37% * 9984 5.760s (1733.3 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.062681 * 9984, metric = 98.37% * 9984 5.760s (1733.3 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.062681 * 9984, metric = 98.37% * 9984 5.760s (1733.3 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.062681 * 9984, metric = 98.37% * 9984 5.760s (1733.3 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.025841 * 9984, metric = 98.08% * 9984 5.831s (1712.2 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.025841 * 9984, metric = 98.08% * 9984 5.831s (1712.2 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.025841 * 9984, metric = 98.08% * 9984 5.831s (1712.2 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.025841 * 9984, metric = 98.08% * 9984 5.831s (1712.2 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.986468 * 9984, metric = 97.76% * 9984 5.255s (1899.9 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.986468 * 9984, metric = 97.76% * 9984 5.255s (1899.9 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.986468 * 9984, metric = 97.76% * 9984 5.255s (1899.9 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.986468 * 9984, metric = 97.76% * 9984 5.255s (1899.9 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.935364 * 10112, metric = 97.48% * 10112 5.344s (1892.2 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.935364 * 10112, metric = 97.48% * 10112 5.344s (1892.2 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.935364 * 10112, metric = 97.48% * 10112 5.344s (1892.2 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.935364 * 10112, metric = 97.48% * 10112 5.344s (1892.2 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.881292 * 9888, metric = 97.17% * 9888 5.745s (1721.1 samples/s);

Finished Epoch[10 of 10]: [Training] loss = 4.881292 * 9888, metric = 97.17% * 9888 5.745s (1721.1 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.881292 * 9888, metric = 97.17% * 9888 5.745s (1721.1 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.881292 * 9888, metric = 97.17% * 9888 5.745s (1721.1 samples/s);
Finished Evaluation [1]: Minibatch[1-79]: metric = 96.53% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 96.53% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 96.53% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 96.53% * 10000;

Appendix F-GPU - Running Aexnet on **8 GPU Nodes**

mpiexec -n 8 --hostfile ./hfile python AlexNet_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 10 --minibatch_size 128
Finished Epoch[1 of 10]: [Training] loss = 5.302910 * 10112, metric = 99.33% * 10112 14.469s (698.9 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.302910 * 10112, metric = 99.33% * 10112 12.968s (779.8 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.302910 * 10112, metric = 99.33% * 10112 12.468s (811.0 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.302910 * 10112, metric = 99.33% * 10112 13.968s (723.9 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.302910 * 10112, metric = 99.33% * 10112 11.968s (844.9 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.302910 * 10112, metric = 99.33% * 10112 13.468s (750.8 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.302910 * 10112, metric = 99.33% * 10112 10.968s (922.0 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.302910 * 10112, metric = 99.33% * 10112 11.468s (881.8 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.283727 * 9984, metric = 99.32% * 9984 4.030s (2477.4 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.283727 * 9984, metric = 99.32% * 9984 4.030s (2477.4 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.283727 * 9984, metric = 99.32% * 9984 4.030s (2477.4 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.283727 * 9984, metric = 99.32% * 9984 4.030s (2477.4 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.283727 * 9984, metric = 99.32% * 9984 4.030s (2477.4 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.283727 * 9984, metric = 99.32% * 9984 4.030s (2477.4 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.283727 * 9984, metric = 99.32% * 9984 4.030s (2477.4 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.283727 * 9984, metric = 99.32% * 9984 4.030s (2477.4 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.183341 * 9984, metric = 99.06% * 9984 3.923s (2545.0 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.183341 * 9984, metric = 99.06% * 9984 3.923s (2545.0 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.183341 * 9984, metric = 99.06% * 9984 3.923s (2545.0 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.183341 * 9984, metric = 99.06% * 9984 3.923s (2545.0 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.183341 * 9984, metric = 99.06% * 9984 3.923s (2545.0 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.183341 * 9984, metric = 99.06% * 9984 3.923s (2545.0 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.183341 * 9984, metric = 99.06% * 9984 3.923s (2545.0 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.183341 * 9984, metric = 99.06% * 9984 3.923s (2545.0 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.132302 * 9984, metric = 98.58% * 9984 3.934s (2537.9 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.132302 * 9984, metric = 98.58% * 9984 3.934s (2537.9 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.132302 * 9984, metric = 98.58% * 9984 3.934s (2537.9 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.132302 * 9984, metric = 98.58% * 9984 3.934s (2537.9 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.132302 * 9984, metric = 98.58% * 9984 3.934s (2537.9 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.132302 * 9984, metric = 98.58% * 9984 3.934s (2537.9 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.132302 * 9984, metric = 98.58% * 9984 3.934s (2537.9 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.132302 * 9984, metric = 98.58% * 9984 3.934s (2537.9 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.099980 * 9984, metric = 98.83% * 9984 4.144s (2409.3 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.099980 * 9984, metric = 98.83% * 9984 4.144s (2409.3 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.099980 * 9984, metric = 98.83% * 9984 4.144s (2409.3 samples/s);

```
Finished Epoch[5 of 10]: [Training] loss = 5.099980 * 9984, metric = 98.83% * 9984 4.144s (2409.3 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.099980 * 9984, metric = 98.83% * 9984 4.144s (2409.3 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.099980 * 9984, metric = 98.83% * 9984 4.144s (2409.3 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.099980 * 9984, metric = 98.83% * 9984 4.144s (2409.3 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.099980 * 9984, metric = 98.83% * 9984 4.144s (2409.3 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.061543 * 9984, metric = 98.33% * 9984 3.906s (2556.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.061543 * 9984, metric = 98.33% * 9984 3.906s (2556.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.061543 * 9984, metric = 98.33% * 9984 3.906s (2556.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.061543 * 9984, metric = 98.33% * 9984 3.906s (2556.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.061543 * 9984, metric = 98.33% * 9984 3.906s (2556.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.061543 * 9984, metric = 98.33% * 9984 3.906s (2556.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.061543 * 9984, metric = 98.33% * 9984 3.906s (2556.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.061543 * 9984, metric = 98.33% * 9984 3.906s (2556.1 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.015182 * 9984, metric = 98.02% * 9984 5.880s (1698.0 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.015182 * 9984, metric = 98.02% * 9984 5.880s (1698.0 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.015182 * 9984, metric = 98.02% * 9984 5.880s (1698.0 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.015182 * 9984, metric = 98.02% * 9984 5.880s (1698.0 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.015182 * 9984, metric = 98.02% * 9984 5.880s (1698.0 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.015182 * 9984, metric = 98.02% * 9984 5.880s (1698.0 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.015182 * 9984, metric = 98.02% * 9984 5.880s (1698.0 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.015182 * 9984, metric = 98.02% * 9984 5.880s (1698.0 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.983348 * 9984, metric = 97.66% * 9984 4.255s (2346.4 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.983348 * 9984, metric = 97.66% * 9984 4.255s (2346.4 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.983348 * 9984, metric = 97.66% * 9984 4.255s (2346.4 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.983348 * 9984, metric = 97.66% * 9984 4.255s (2346.4 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.983348 * 9984, metric = 97.66% * 9984 4.255s (2346.4 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.983348 * 9984, metric = 97.66% * 9984 4.255s (2346.4 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.983348 * 9984, metric = 97.66% * 9984 4.255s (2346.4 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 4.983348 * 9984, metric = 97.66% * 9984 4.255s (2346.4 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.939767 * 10112, metric = 97.26% * 10112 6.789s (1489.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.939767 * 10112, metric = 97.26% * 10112 6.789s (1489.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.939767 * 10112, metric = 97.26% * 10112 6.789s (1489.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.939767 * 10112, metric = 97.26% * 10112 6.789s (1489.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.939767 * 10112, metric = 97.26% * 10112 6.789s (1489.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.939767 * 10112, metric = 97.26% * 10112 6.789s (1489.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.939767 * 10112, metric = 97.26% * 10112 6.789s (1489.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 4.939767 * 10112, metric = 97.26% * 10112 6.789s (1489.5 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.904487 * 9888, metric = 97.09% * 9888 4.257s (2322.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.904487 * 9888, metric = 97.09% * 9888 4.257s (2322.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.904487 * 9888, metric = 97.09% * 9888 4.257s (2322.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.904487 * 9888, metric = 97.09% * 9888 4.257s (2322.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.904487 * 9888, metric = 97.09% * 9888 4.257s (2322.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.904487 * 9888, metric = 97.09% * 9888 4.257s (2322.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.904487 * 9888, metric = 97.09% * 9888 4.257s (2322.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 4.904487 * 9888, metric = 97.09% * 9888 4.257s (2322.8 samples/s);
Finished Evaluation [1]: Minibatch[1-79]: metric = 97.15% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 97.15% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 97.15% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 97.15% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 97.15% * 10000;
```

Finished Evaluation [1]: Minibatch[1-79]: metric = 97.15% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 97.15% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 97.15% * 10000;

Appendix G-GPU - Running Resnet on **GPU** with 10k epoch size, 256 batch size, single node

python TrainResNet_ImageNet_Distributed.py --network resnet34 --epochs 10 --epoch_size 1000

Selected GPU[0] Tesla P100-PCIE-16GB as the process wide default device.

Finished Epoch[1 of 10]: [Training] loss = 6.217681 * 1024, metric = 98.93% * 1024 15.718s (65.1 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.751491 * 992, metric = 98.79% * 992 5.123s (193.6 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.632725 * 992, metric = 98.59% * 992 5.257s (188.7 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.535466 * 992, metric = 99.40% * 992 5.391s (184.0 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.450279 * 1024, metric = 99.12% * 1024 5.290s (193.6 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.412857 * 992, metric = 99.50% * 992 5.068s (195.7 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.367375 * 992, metric = 98.69% * 992 5.136s (193.1 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.314719 * 992, metric = 98.39% * 992 5.091s (194.9 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.348723 * 1024, metric = 98.63% * 1024 5.348s (191.5 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.280454 * 976, metric = 98.98% * 976 5.065s (192.7 samples/s);

Finished Evaluation [1]: Minibatch[1-313]: metric = 98.55% * 10000;

Appendix H-GPU - Resnet on 2 GPU Nodes (Default)

mpiexec -n 2 --hostfile ./hfile python TrainResNet_ImageNet_Distributed.py --network resnet34 --epochs 10 --epoch_size 1000
Finished Epoch[1 of 10]: [Training] loss = 6.258869 * 1024, metric = 98.83% * 1024 15.273s (67.0 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 6.258869 * 1024, metric = 98.83% * 1024 14.773s (69.3 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.720713 * 1024, metric = 99.12% * 1024 4.572s (224.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.720713 * 1024, metric = 99.12% * 1024 4.572s (224.0 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.570447 * 960, metric = 98.96% * 960 4.358s (220.3 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.570447 * 960, metric = 98.96% * 960 4.358s (220.3 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.510996 * 1024, metric = 99.12% * 1024 4.409s (232.3 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.510996 * 1024, metric = 99.12% * 1024 4.409s (232.3 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.407166 * 1024, metric = 99.22% * 1024 4.587s (223.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.407166 * 1024, metric = 99.22% * 1024 4.587s (223.2 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.382329 * 960, metric = 99.06% * 960 4.642s (206.8 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.382329 * 960, metric = 99.06% * 960 4.642s (206.8 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.391663 * 1024, metric = 98.93% * 1024 4.508s (227.2 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.391663 * 1024, metric = 98.93% * 1024 4.509s (227.1 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.303835 * 960, metric = 98.33% * 960 4.626s (207.5 samples/s);

Finished Epoch[8 of 10]: [Training] loss = 5.303835 * 960, metric = 98.33% * 960 4.626s (207.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.313035 * 1024, metric = 98.14% * 1024 4.669s (219.3 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.313035 * 1024, metric = 98.14% * 1024 4.669s (219.3 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.262076 * 976, metric = 98.57% * 976 4.522s (215.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.262076 * 976, metric = 98.57% * 976 4.522s (215.8 samples/s);
Finished Evaluation [1]: Minibatch[1-157]: metric = 99.23% * 10000;
Finished Evaluation [1]: Minibatch[1-157]: metric = 99.23% * 10000;

Appendix I-GPU - Resnet on 4 GPU Nodes (Default)

(local-cntk-gpu) [kshafie@o0768 HiDL]$ mpiexec -n 4 --hostfile ./hostfile python TrainResNet_ImageNet_Distributed.py --network resnet34 --epochs 10 --epoch_size 1000
Finished Epoch[1 of 10]: [Training] loss = 6.347377 * 1024, metric = 99.32% * 1024 14.003s (73.1 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 6.347377 * 1024, metric = 99.32% * 1024 13.503s (75.8 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 6.347377 * 1024, metric = 99.32% * 1024 13.003s (78.8 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 6.347377 * 1024, metric = 99.32% * 1024 12.503s (81.9 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.712100 * 1024, metric = 99.51% * 1024 3.478s (294.4 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.712100 * 1024, metric = 99.51% * 1024 3.478s (294.4 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.712100 * 1024, metric = 99.51% * 1024 3.478s (294.4 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.712100 * 1024, metric = 99.51% * 1024 3.478s (294.4 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.568633 * 1024, metric = 99.32% * 1024 3.375s (303.4 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.568633 * 1024, metric = 99.32% * 1024 3.374s (303.5 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.568633 * 1024, metric = 99.32% * 1024 3.375s (303.4 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.568633 * 1024, metric = 99.32% * 1024 3.375s (303.4 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.440408 * 1024, metric = 98.73% * 1024 3.370s (303.9 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.440408 * 1024, metric = 98.73% * 1024 3.371s (303.8 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.440408 * 1024, metric = 98.73% * 1024 3.371s (303.8 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.440408 * 1024, metric = 98.73% * 1024 3.371s (303.8 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.366688 * 1024, metric = 98.54% * 1024 3.411s (300.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.366688 * 1024, metric = 98.54% * 1024 3.411s (300.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.366688 * 1024, metric = 98.54% * 1024 3.411s (300.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.366688 * 1024, metric = 98.54% * 1024 3.411s (300.2 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.348842 * 896, metric = 99.11% * 896 3.289s (272.4 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.348842 * 896, metric = 99.11% * 896 3.289s (272.4 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.348842 * 896, metric = 99.11% * 896 3.289s (272.4 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.348842 * 896, metric = 99.11% * 896 3.289s (272.4 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.322204 * 1024, metric = 98.24% * 1024 3.490s (293.4 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.322204 * 1024, metric = 98.24% * 1024 3.490s (293.4 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.322204 * 1024, metric = 98.24% * 1024 3.490s (293.4 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.322204 * 1024, metric = 98.24% * 1024 3.490s (293.4 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.277085 * 1024, metric = 98.34% * 1024 3.372s (303.7 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.277085 * 1024, metric = 98.34% * 1024 3.372s (303.7 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.277085 * 1024, metric = 98.34% * 1024 3.372s (303.7 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.277085 * 1024, metric = 98.34% * 1024 3.372s (303.7 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.220421 * 1024, metric = 98.24% * 1024 3.389s (302.2 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.220421 * 1024, metric = 98.24% * 1024 3.389s (302.2 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.220421 * 1024, metric = 98.24% * 1024 3.389s (302.2 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.220421 * 1024, metric = 98.24% * 1024 3.389s (302.2 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.233425 * 912, metric = 97.59% * 912 3.335s (273.5 samples/s);

Finished Epoch[10 of 10]: [Training] loss = 5.233425 * 912, metric = 97.59% * 912 3.335s (273.5 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.233425 * 912, metric = 97.59% * 912 3.335s (273.5 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.233425 * 912, metric = 97.59% * 912 3.335s (273.5 samples/s);
Finished Evaluation [1]: Minibatch[1-79]: metric = 99.52% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 99.52% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 99.52% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 99.52% * 10000;

Appendix J-GPU - Resnet on 8 GPU Nodes (Default)

(local-cntk-gpu) [kshafie@o0767 HiDL]$ mpiexec -n 8 --hostfile ./hfile python TrainResNet_ImageNet_Distributed.py --network resnet34 --epochs 10 --epoch_size 1000
Finished Epoch[1 of 10]: [Training] loss = 6.475858 * 1024, metric = 99.02% * 1024 12.831s (79.8 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 6.475858 * 1024, metric = 99.02% * 1024 12.331s (83.0 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 6.475858 * 1024, metric = 99.02% * 1024 13.331s (76.8 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 6.475858 * 1024, metric = 99.02% * 1024 15.331s (66.8 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 6.475858 * 1024, metric = 99.02% * 1024 14.331s (71.5 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 6.475858 * 1024, metric = 99.02% * 1024 14.831s (69.0 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 6.475858 * 1024, metric = 99.02% * 1024 15.831s (64.7 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 6.475858 * 1024, metric = 99.02% * 1024 13.831s (74.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.958034 * 1024, metric = 99.12% * 1024 3.048s (336.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.958034 * 1024, metric = 99.12% * 1024 3.048s (336.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.958034 * 1024, metric = 99.12% * 1024 3.048s (336.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.958034 * 1024, metric = 99.12% * 1024 3.048s (336.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.958034 * 1024, metric = 99.12% * 1024 3.048s (336.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.958034 * 1024, metric = 99.12% * 1024 3.048s (336.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.958034 * 1024, metric = 99.12% * 1024 3.049s (335.8 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.958034 * 1024, metric = 99.12% * 1024 3.048s (336.0 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.917967 * 1024, metric = 99.32% * 1024 3.059s (334.7 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.917967 * 1024, metric = 99.32% * 1024 3.059s (334.7 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.917967 * 1024, metric = 99.32% * 1024 3.059s (334.7 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.917967 * 1024, metric = 99.32% * 1024 3.060s (334.6 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.917967 * 1024, metric = 99.32% * 1024 3.060s (334.6 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.917967 * 1024, metric = 99.32% * 1024 3.060s (334.6 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.917967 * 1024, metric = 99.32% * 1024 3.060s (334.6 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.917967 * 1024, metric = 99.32% * 1024 3.060s (334.6 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.763921 * 1024, metric = 99.02% * 1024 3.061s (334.5 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.763921 * 1024, metric = 99.02% * 1024 3.061s (334.5 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.763921 * 1024, metric = 99.02% * 1024 3.061s (334.5 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.763921 * 1024, metric = 99.02% * 1024 3.061s (334.5 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.763921 * 1024, metric = 99.02% * 1024 3.061s (334.5 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.763921 * 1024, metric = 99.02% * 1024 3.061s (334.5 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.763921 * 1024, metric = 99.02% * 1024 3.061s (334.5 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.760674 * 1024, metric = 99.02% * 1024 2.983s (343.3 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.760674 * 1024, metric = 99.02% * 1024 2.984s (343.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.760674 * 1024, metric = 99.02% * 1024 2.983s (343.3 samples/s);

```
Finished Epoch[5 of 10]: [Training] loss = 5.760674 * 1024, metric = 99.02% * 1024 2.984s (343.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.760674 * 1024, metric = 99.02% * 1024 2.984s (343.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.760674 * 1024, metric = 99.02% * 1024 2.984s (343.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.760674 * 1024, metric = 99.02% * 1024 2.984s (343.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.760674 * 1024, metric = 99.02% * 1024 2.984s (343.2 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.689043 * 1024, metric = 98.73% * 1024 2.989s (342.6 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.689043 * 1024, metric = 98.73% * 1024 2.989s (342.6 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.689043 * 1024, metric = 98.73% * 1024 2.989s (342.6 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.689043 * 1024, metric = 98.73% * 1024 2.989s (342.6 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.689043 * 1024, metric = 98.73% * 1024 2.989s (342.6 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.689043 * 1024, metric = 98.73% * 1024 2.989s (342.6 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.689043 * 1024, metric = 98.73% * 1024 2.989s (342.6 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.689043 * 1024, metric = 98.73% * 1024 2.990s (342.5 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.580080 * 1024, metric = 98.93% * 1024 2.972s (344.5 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.580080 * 1024, metric = 98.93% * 1024 2.973s (344.4 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.580080 * 1024, metric = 98.93% * 1024 2.973s (344.4 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.580080 * 1024, metric = 98.93% * 1024 2.973s (344.4 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.580080 * 1024, metric = 98.93% * 1024 2.973s (344.4 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.580080 * 1024, metric = 98.93% * 1024 2.973s (344.4 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.580080 * 1024, metric = 98.93% * 1024 2.973s (344.4 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.580080 * 1024, metric = 98.93% * 1024 2.973s (344.4 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.367925 * 1024, metric = 98.73% * 1024 2.993s (342.1 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.367925 * 1024, metric = 98.73% * 1024 2.993s (342.1 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.367925 * 1024, metric = 98.73% * 1024 2.993s (342.1 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.367925 * 1024, metric = 98.73% * 1024 2.993s (342.1 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.367925 * 1024, metric = 98.73% * 1024 2.993s (342.1 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.367925 * 1024, metric = 98.73% * 1024 2.993s (342.1 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.367925 * 1024, metric = 98.73% * 1024 2.993s (342.1 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.367925 * 1024, metric = 98.73% * 1024 2.993s (342.1 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.381828 * 1024, metric = 98.63% * 1024 3.053s (335.4 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.381828 * 1024, metric = 98.63% * 1024 3.053s (335.4 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.381828 * 1024, metric = 98.63% * 1024 3.053s (335.4 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.381828 * 1024, metric = 98.63% * 1024 3.053s (335.4 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.381828 * 1024, metric = 98.63% * 1024 3.053s (335.4 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.381828 * 1024, metric = 98.63% * 1024 3.054s (335.3 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.381828 * 1024, metric = 98.63% * 1024 3.053s (335.4 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.381828 * 1024, metric = 98.63% * 1024 3.054s (335.3 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.437615 * 784, metric = 98.47% * 784 3.006s (260.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.437615 * 784, metric = 98.47% * 784 3.007s (260.7 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.437615 * 784, metric = 98.47% * 784 3.006s (260.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.437615 * 784, metric = 98.47% * 784 3.006s (260.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.437615 * 784, metric = 98.47% * 784 3.006s (260.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.437615 * 784, metric = 98.47% * 784 3.006s (260.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.437615 * 784, metric = 98.47% * 784 3.007s (260.7 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.437615 * 784, metric = 98.47% * 784 3.007s (260.7 samples/s);
Finished Evaluation [1]: Minibatch[1-40]: metric = 99.50% * 10000;
Finished Evaluation [1]: Minibatch[1-40]: metric = 99.50% * 10000;
Finished Evaluation [1]: Minibatch[1-40]: metric = 99.50% * 10000;
Finished Evaluation [1]: Minibatch[1-40]: metric = 99.50% * 10000;
Finished Evaluation [1]: Minibatch[1-40]: metric = 99.50% * 10000;
```

Finished Evaluation [1]: Minibatch[1-40]: metric = 99.50% * 10000;
Finished Evaluation [1]: Minibatch[1-40]: metric = 99.50% * 10000;
Finished Evaluation [1]: Minibatch[1-40]: metric = 99.50% * 10000;

Appendix K-GPU - VGG16 on 1 GPU Node

(local-cntk-gpu) [kshafie@o0756 HiDL]$ python VGG16_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 10 --minibatch_size 128
Training 40708104 parameters in 32 parameter tensors.

Finished Epoch[1 of 10]: [Training] loss = 5.298573 * 10112, metric = 99.39% * 10112 31.008s (326.1 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.284860 * 9984, metric = 99.51% * 9984 19.897s (501.8 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.261325 * 9984, metric = 99.25% * 9984 19.948s (500.5 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.242877 * 9984, metric = 99.19% * 9984 20.184s (494.6 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.244332 * 9984, metric = 99.29% * 9984 19.459s (513.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.239106 * 9984, metric = 99.20% * 9984 20.883s (478.1 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.217385 * 9984, metric = 99.00% * 9984 20.150s (495.5 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.213807 * 9984, metric = 99.06% * 9984 21.050s (474.3 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.227511 * 10112, metric = 99.10% * 10112 19.313s (523.6 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.203870 * 9888, metric = 99.05% * 9888 20.730s (477.0 samples/s);
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.84% * 10000;

Appendix L-GPU - VGG16 on 2 GPU Nodes

(local-cntk-gpu) [kshafie@o0768 HiDL]$ mpiexec -n 2 --hostfile ./hostfile python VGG16_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 10 --minibatch_size 128
Training 40708104 parameters in 32 parameter tensors.

Finished Epoch[1 of 10]: [Training] loss = 5.296609 * 10112, metric = 99.49% * 10112 17.672s (572.2 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.296609 * 10112, metric = 99.49% * 10112 17.172s (588.9 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.267974 * 9984, metric = 99.42% * 9984 12.545s (795.9 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.267974 * 9984, metric = 99.42% * 9984 12.547s (795.7 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.252537 * 9984, metric = 99.12% * 9984 12.984s (768.9 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.252537 * 9984, metric = 99.12% * 9984 12.983s (769.0 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.236353 * 9984, metric = 98.98% * 9984 12.582s (793.5 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.236353 * 9984, metric = 98.98% * 9984 12.587s (793.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.243788 * 9984, metric = 99.28% * 9984 12.721s (784.8 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.243788 * 9984, metric = 99.28% * 9984 12.721s (784.8 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.229681 * 9984, metric = 99.08% * 9984 12.576s (793.9 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.229681 * 9984, metric = 99.08% * 9984 12.576s (793.9 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.216612 * 9984, metric = 99.11% * 9984 12.835s (777.9 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.216612 * 9984, metric = 99.11% * 9984 12.841s (777.5 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.212117 * 9984, metric = 99.04% * 9984 12.868s (775.9 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.212117 * 9984, metric = 99.04% * 9984 12.859s (776.4 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.218119 * 10112, metric = 99.09% * 10112 15.187s (665.8 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.218119 * 10112, metric = 99.09% * 10112 15.184s (666.0 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.202665 * 9888, metric = 98.89% * 9888 14.289s (692.0 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.202665 * 9888, metric = 98.89% * 9888 14.296s (691.7 samples/s);

Finished Evaluation [1]: Minibatch[1-79]: metric = 98.96% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.96% * 10000;

Appendix M-GPU - VGG16 on 4 GPU Nodes

(local-cntk-gpu) [kshafie@o0768 HiDL]$ mpiexec -n 4 --hostfile ./hostfile python VGG16_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 10 --minibatch_size 128
Finished Epoch[1 of 10]: [Training] loss = 5.297753 * 10112, metric = 99.35% * 10112 11.512s (878.4 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.297753 * 10112, metric = 99.35% * 10112 10.512s (961.9 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.297753 * 10112, metric = 99.35% * 10112 11.012s (918.3 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.297753 * 10112, metric = 99.35% * 10112 10.012s (1010.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.278848 * 9984, metric = 99.34% * 9984 5.462s (1827.9 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.278848 * 9984, metric = 99.34% * 9984 5.462s (1827.9 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.278848 * 9984, metric = 99.34% * 9984 5.462s (1827.9 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.278848 * 9984, metric = 99.34% * 9984 5.462s (1827.9 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.261238 * 9984, metric = 99.23% * 9984 5.555s (1797.3 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.261238 * 9984, metric = 99.23% * 9984 5.555s (1797.3 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.261238 * 9984, metric = 99.23% * 9984 5.555s (1797.3 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.261238 * 9984, metric = 99.23% * 9984 5.555s (1797.3 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.241182 * 9984, metric = 99.16% * 9984 5.754s (1735.1 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.241182 * 9984, metric = 99.16% * 9984 5.754s (1735.1 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.241182 * 9984, metric = 99.16% * 9984 5.754s (1735.1 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.241182 * 9984, metric = 99.16% * 9984 5.754s (1735.1 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.241886 * 9984, metric = 99.29% * 9984 5.635s (1771.8 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.241886 * 9984, metric = 99.29% * 9984 5.635s (1771.8 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.241886 * 9984, metric = 99.29% * 9984 5.635s (1771.8 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.241886 * 9984, metric = 99.29% * 9984 5.635s (1771.8 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.230725 * 9984, metric = 99.19% * 9984 6.277s (1590.6 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.230725 * 9984, metric = 99.19% * 9984 6.277s (1590.6 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.230725 * 9984, metric = 99.19% * 9984 6.277s (1590.6 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.230725 * 9984, metric = 99.19% * 9984 6.277s (1590.6 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.219468 * 9984, metric = 98.96% * 9984 5.823s (1714.6 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.219468 * 9984, metric = 98.96% * 9984 5.823s (1714.6 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.219468 * 9984, metric = 98.96% * 9984 5.823s (1714.6 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.219468 * 9984, metric = 98.96% * 9984 5.824s (1714.3 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.211505 * 9984, metric = 99.08% * 9984 6.025s (1657.1 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.211505 * 9984, metric = 99.08% * 9984 6.026s (1656.8 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.211505 * 9984, metric = 99.08% * 9984 6.026s (1656.8 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.211505 * 9984, metric = 99.08% * 9984 6.026s (1656.8 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.213564 * 10112, metric = 99.14% * 10112 5.836s (1732.7 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.213564 * 10112, metric = 99.14% * 10112 5.836s (1732.7 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.213564 * 10112, metric = 99.14% * 10112 5.836s (1732.7 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.213564 * 10112, metric = 99.14% * 10112 5.836s (1732.7 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.202469 * 9888, metric = 98.94% * 9888 5.515s (1792.9 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.202469 * 9888, metric = 98.94% * 9888 5.515s (1792.9 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.202469 * 9888, metric = 98.94% * 9888 5.515s (1792.9 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.202469 * 9888, metric = 98.94% * 9888 5.515s (1792.9 samples/s);
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.96% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.96% * 10000;

Finished Evaluation [1]: Minibatch[1-79]: metric = 98.96% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.96% * 10000;

Appendix N-GPU - VGG16 on 8 GPU Nodes

(local-cntk-gpu) [kshafie@o0767 HiDL]$ mpiexec -n 8 --hostfile ./hfile python VGG16_ImageNet_Distributed.py --epoch_size 10000 --num_epochs 10 --minibatch_size 128
Selected GPU[0] Tesla P100-PCIE-16GB as the process wide default device.
Training 40708104 parameters in 32 parameter tensors.
Training 40708104 parameters in 32 parameter tensors.

Finished Epoch[1 of 10]: [Training] loss = 5.298381 * 10112, metric = 99.35% * 10112 13.418s (753.6 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.298381 * 10112, metric = 99.35% * 10112 9.917s (1019.7 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.298381 * 10112, metric = 99.35% * 10112 12.418s (814.3 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.298381 * 10112, metric = 99.35% * 10112 10.918s (926.2 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.298381 * 10112, metric = 99.35% * 10112 12.917s (782.8 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.298381 * 10112, metric = 99.35% * 10112 11.418s (885.6 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.298381 * 10112, metric = 99.35% * 10112 11.918s (848.5 samples/s);
Finished Epoch[1 of 10]: [Training] loss = 5.298381 * 10112, metric = 99.35% * 10112 10.417s (970.7 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.285533 * 9984, metric = 99.42% * 9984 5.291s (1887.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.285533 * 9984, metric = 99.42% * 9984 5.291s (1887.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.285533 * 9984, metric = 99.42% * 9984 5.291s (1887.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.285533 * 9984, metric = 99.42% * 9984 5.291s (1887.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.285533 * 9984, metric = 99.42% * 9984 5.291s (1887.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.285533 * 9984, metric = 99.42% * 9984 5.291s (1887.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.285533 * 9984, metric = 99.42% * 9984 5.291s (1887.0 samples/s);
Finished Epoch[2 of 10]: [Training] loss = 5.285533 * 9984, metric = 99.42% * 9984 5.291s (1887.0 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.266152 * 9984, metric = 99.25% * 9984 5.568s (1793.1 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.266152 * 9984, metric = 99.25% * 9984 5.568s (1793.1 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.266152 * 9984, metric = 99.25% * 9984 5.568s (1793.1 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.266152 * 9984, metric = 99.25% * 9984 5.568s (1793.1 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.266152 * 9984, metric = 99.25% * 9984 5.568s (1793.1 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.266152 * 9984, metric = 99.25% * 9984 5.568s (1793.1 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.266152 * 9984, metric = 99.25% * 9984 5.568s (1793.1 samples/s);
Finished Epoch[3 of 10]: [Training] loss = 5.266152 * 9984, metric = 99.25% * 9984 5.568s (1793.1 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.243866 * 9984, metric = 99.11% * 9984 5.295s (1885.6 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.243866 * 9984, metric = 99.11% * 9984 5.294s (1885.9 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.243866 * 9984, metric = 99.11% * 9984 5.295s (1885.6 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.243866 * 9984, metric = 99.11% * 9984 5.295s (1885.6 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.243866 * 9984, metric = 99.11% * 9984 5.295s (1885.6 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.243866 * 9984, metric = 99.11% * 9984 5.295s (1885.6 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.243866 * 9984, metric = 99.11% * 9984 5.295s (1885.6 samples/s);
Finished Epoch[4 of 10]: [Training] loss = 5.243866 * 9984, metric = 99.11% * 9984 5.295s (1885.6 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.243922 * 9984, metric = 99.24% * 9984 5.327s (1874.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.243922 * 9984, metric = 99.24% * 9984 5.327s (1874.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.243922 * 9984, metric = 99.24% * 9984 5.327s (1874.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.243922 * 9984, metric = 99.24% * 9984 5.327s (1874.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.243922 * 9984, metric = 99.24% * 9984 5.327s (1874.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.243922 * 9984, metric = 99.24% * 9984 5.327s (1874.2 samples/s);

```
Finished Epoch[5 of 10]: [Training] loss = 5.243922 * 9984, metric = 99.24% * 9984 5.327s (1874.2 samples/s);
Finished Epoch[5 of 10]: [Training] loss = 5.243922 * 9984, metric = 99.24% * 9984 5.327s (1874.2 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.232617 * 9984, metric = 99.31% * 9984 5.333s (1872.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.232617 * 9984, metric = 99.31% * 9984 5.333s (1872.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.232617 * 9984, metric = 99.31% * 9984 5.333s (1872.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.232617 * 9984, metric = 99.31% * 9984 5.333s (1872.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.232617 * 9984, metric = 99.31% * 9984 5.333s (1872.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.232617 * 9984, metric = 99.31% * 9984 5.333s (1872.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.232617 * 9984, metric = 99.31% * 9984 5.333s (1872.1 samples/s);
Finished Epoch[6 of 10]: [Training] loss = 5.232617 * 9984, metric = 99.31% * 9984 5.333s (1872.1 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.216215 * 9984, metric = 99.14% * 9984 5.424s (1840.7 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.216215 * 9984, metric = 99.14% * 9984 5.424s (1840.7 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.216215 * 9984, metric = 99.14% * 9984 5.424s (1840.7 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.216215 * 9984, metric = 99.14% * 9984 5.424s (1840.7 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.216215 * 9984, metric = 99.14% * 9984 5.424s (1840.7 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.216215 * 9984, metric = 99.14% * 9984 5.424s (1840.7 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.216215 * 9984, metric = 99.14% * 9984 5.424s (1840.7 samples/s);
Finished Epoch[7 of 10]: [Training] loss = 5.216215 * 9984, metric = 99.14% * 9984 5.424s (1840.7 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.213129 * 9984, metric = 99.04% * 9984 5.282s (1890.2 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.213129 * 9984, metric = 99.04% * 9984 5.282s (1890.2 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.213129 * 9984, metric = 99.04% * 9984 5.282s (1890.2 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.213129 * 9984, metric = 99.04% * 9984 5.281s (1890.6 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.213129 * 9984, metric = 99.04% * 9984 5.282s (1890.2 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.213129 * 9984, metric = 99.04% * 9984 5.282s (1890.2 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.213129 * 9984, metric = 99.04% * 9984 5.282s (1890.2 samples/s);
Finished Epoch[8 of 10]: [Training] loss = 5.213129 * 9984, metric = 99.04% * 9984 5.282s (1890.2 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.215731 * 10112, metric = 99.00% * 10112 5.363s (1885.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.215731 * 10112, metric = 99.00% * 10112 5.363s (1885.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.215731 * 10112, metric = 99.00% * 10112 5.363s (1885.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.215731 * 10112, metric = 99.00% * 10112 5.363s (1885.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.215731 * 10112, metric = 99.00% * 10112 5.363s (1885.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.215731 * 10112, metric = 99.00% * 10112 5.363s (1885.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.215731 * 10112, metric = 99.00% * 10112 5.363s (1885.5 samples/s);
Finished Epoch[9 of 10]: [Training] loss = 5.215731 * 10112, metric = 99.00% * 10112 5.363s (1885.5 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.203700 * 9888, metric = 98.94% * 9888 5.311s (1861.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.203700 * 9888, metric = 98.94% * 9888 5.311s (1861.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.203700 * 9888, metric = 98.94% * 9888 5.311s (1861.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.203700 * 9888, metric = 98.94% * 9888 5.311s (1861.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.203700 * 9888, metric = 98.94% * 9888 5.311s (1861.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.203700 * 9888, metric = 98.94% * 9888 5.311s (1861.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.203700 * 9888, metric = 98.94% * 9888 5.311s (1861.8 samples/s);
Finished Epoch[10 of 10]: [Training] loss = 5.203700 * 9888, metric = 98.94% * 9888 5.311s (1861.8 samples/s);
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.93% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.93% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.93% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.93% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.93% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.93% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.93% * 10000;
Finished Evaluation [1]: Minibatch[1-79]: metric = 98.93% * 10000;
```