

## Lab 11: Distributed TensorFlow using Horovod

*Authors: Arias Rodrigo, Burca Horia*

*Date: 31/12/2018*

The Horovod framework is analyzed to measure the performance of training the CIFAR10 dataset, using different models.

### About the performance measurements

In the original post, the time counters used were the ones reported by Horovod, using the hooks available in TensorFlow.

We included another metric, which consists of computing the wall time of the training function. The time is divided in the initialization time, and the actual training time. The former is not considered for computing the speedup.

With the maximum number of steps set to 500, we measure the following for the `resnet_v2_101` model:

GPUs	Init time(s)	Train time(s)	Total time(s)	Speedup (train)
1	68	163	231	1.00
2	142	93	235	1.72
4	75	47	122	3.47
8	110	30	140	5.43
16	110	19	129	8.58
32	111	14	125	11.64

We can see that the speedup is not very close to linear. Also, the overhead introduced by the large initialization time is very large for those small train samples.

If we take a look at the hook `average_example_per_sec`, we can see the following values:

GPUs	Examples/s	Speedup
1	202	1.00
2	191	1.89
4	193	3.82
8	191	7.56
16	193	15.29
32	190	30.09

Which may seem far superior to the values stated in the previous table.

## Models tested

We initially tested the same model analyzed in the post, `resnet_v2_101`, in order to verify our execution. And we got very similar results when using the same performance metric.

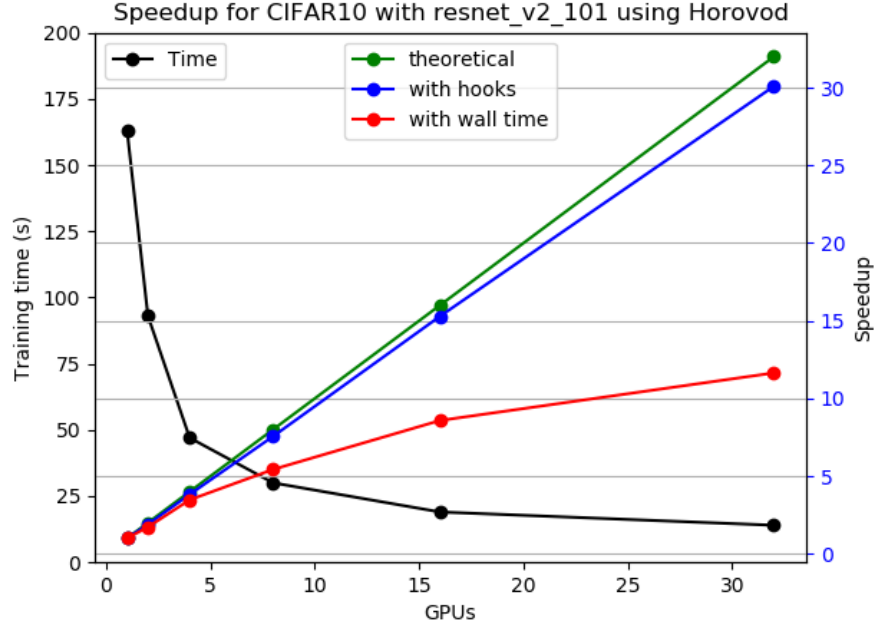


Figure 1: Speedup plot for `resnet_v2_101`

We can observe how the training time is decreased as we increment the number of processing units. Also the two measurements of the speedup differ as the number of GPUs grow.

The model `vgg_19` with the same `max_step` parameter, set to 500 is also analyzed. We observe a better speedup, with both metrics.

## Conclusions

The measurement produced by the `average_example_per_sec` hooks in TensorFlow, leads to different time, compared with the wall time of the train process. We assume that the difference is due to overheads, such as the initialization or communication time.

A more in depth analysis could provide a better explanation on why such differences are observed, and which metric is more reliable to represent the speedup of the training process.

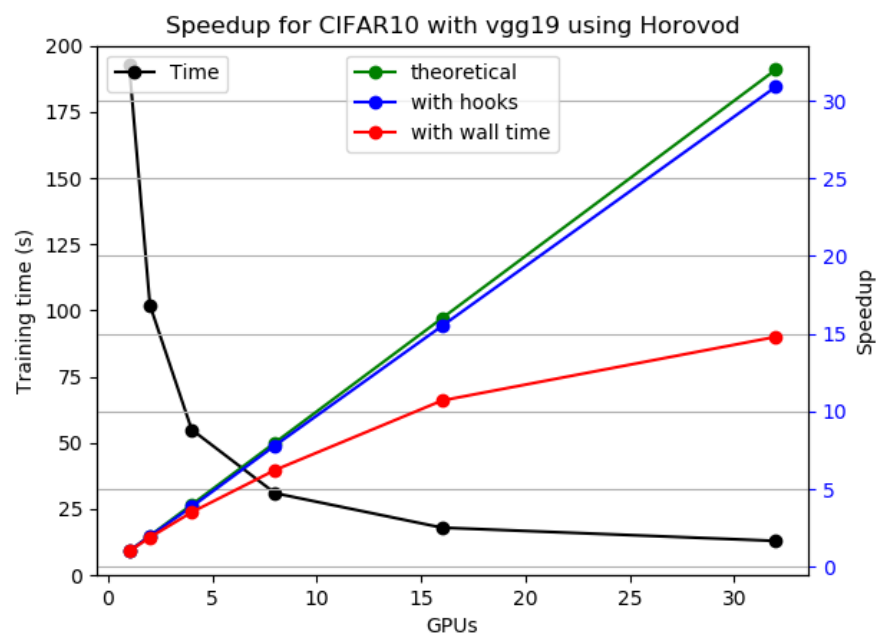


Figure 2: Speedup plot for vgg19