

Horovod

Horovod 0.15.2

Na računalnom klasteru Isabella, na čvorovima s grafičkim procesorima NVIDIA Tesla V100-SXM2-16GB, instaliran je Horovod, Pythonova biblioteka za distribuirano pokretanje TensorFlow i PyTorch poslova.

Horovod je instaliran za Python 3.5, uz sljedeće verzije NVIDIA alata i biblioteka za strojno učenje:

- CUDA 10.0
- cuDNN 7.3.1
- NCCL 2.3.5

Korištenje

Horovod zahtijeva minimalne izmjene Python skripti napisanih za izvođenje na jednom grafičkom procesoru i omogućava njihovo izvođenje na više grafičkih procesora na jednom ili više radnih čvorova klastera.



Važno

Za sve TensorFlow i PyTorch poslove koji zahtijevaju **više grafičkih procesora**, **obavezno** je korištenje biblioteke **Horovod**!

Primjer korištenja biblioteke Horovod:

```
import tensorflow as tf
import horovod.tensorflow as hvd

# Initialize Horovod
hvd.init()

# Pin GPU to be used to process local rank (one GPU per process)
config = tf.ConfigProto()
config.gpu_options.visible_device_list = str(hvd.local_rank())

# Build model...
loss = ...
opt = tf.train.AdagradOptimizer(0.01 * hvd.size())

# Add Horovod Distributed Optimizer
opt = hvd.DistributedOptimizer(opt)

# Add hook to broadcast variables from rank 0 to all other processes during
# initialization.
hooks = [hvd.BroadcastGlobalVariablesHook(0)]

# Make training operation
train_op = opt.minimize(loss)

# Save checkpoints only on worker 0 to prevent other workers from corrupting them.
checkpoint_dir = '/tmp/train_logs' if hvd.rank() == 0 else None

# The MonitoredTrainingSession takes care of session initialization,
# restoring from a checkpoint, saving to a checkpoint, and closing when done
# or an error occurs.
with tf.train.MonitoredTrainingSession(checkpoint_dir=checkpoint_dir,
                                       config=config,
                                       hooks=hooks) as mon_sess:
    while not mon_sess.should_stop():
        # Perform synchronous training.
        mon_sess.run(train_op)
```

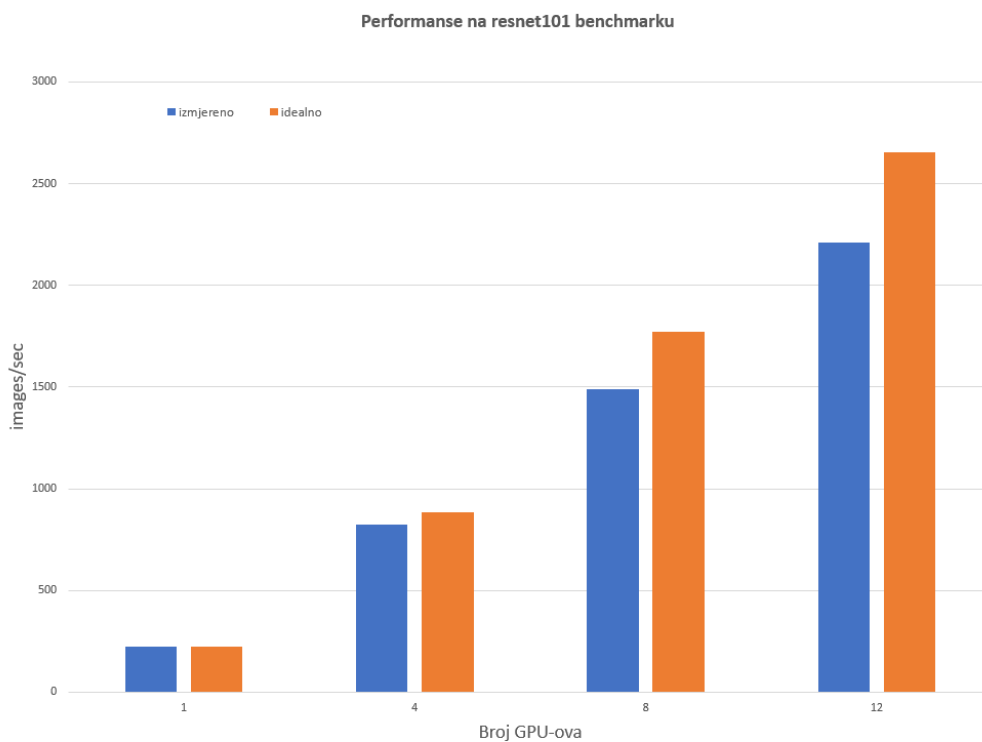


Više informacija o korištenju biblioteke Horovod, kao i cjelovite primjere TensorFlow i PyTorch skripti, možete pronaći na [Horovod GitHub repozitoriju](#).

Performanse

Performanse paralelnog izvođenja TensorFlow aplikacija korištenjem biblioteke Horovod mjerene su standardnim *resnet101* benchmarkom iz službenog TensorFlow [benchmark repozitorija](#).

Slika 1 prikazuje performanse na *resnet101* testu u odnosu na broj GPU-ova. Vidljivo je da se s povećanjem broja GPU-ova povećava gubitak u odnosu na maksimalne teoretske performanse, što je i za očekivati. Međutim, čak i za 12 GPU-ova, sustav daje oko 85% idealnog slučaja. Idealne performanse procijenjene su iz rezultata istog benchmarka na jednom grafičkom procesoru, bez korištenja biblioteke Horovod.



Slika 2 prikazuje performanse na istom benchmarku, korištenjem 4 GPU-a u različitim raspodjelama po čvorovima. Vidljivo je da raspored GPU-ova po čvorovima ne utječe bitno na performanse na ovom benchmarku.

Performanse na resnet101 benchmarku - usporedba
raspodjele GPU-ova po čvorovima

