



Chapter 5.3: Running DNN Multi-GPU Training On ROCm

ROCm Tutorial | AMD 2020

Table of Contents

CHAPTER 5.3: RUNNING DNN MULTI-GPU TRAINING ON ROCM.....	2
PREPARATION.....	2
RUNNING THE EXAMPLE.....	3

Chapter 5.3: Running DNN Multi-GPU Training On ROCm

This hands-on tutorial shows how we can get started with the popular Keras framework and deploy a MNIST model that uses multiple GPUs on ROCm.

Preparation

1. We will be using the AMD provided Tensorflow docker container for this tutorial
 - Please ensure that ROCm is correctly installed and docker is also set up using the instructions in our installation guide
2. Obtain the TensorFlow docker image:
 - `docker pull rocm/tensorflow`
3. Start the docker container:
 - `alias drun='sudo docker run -it --network=host --device=/dev/kfd --device=/dev/dri --ipc=host --shm-size 16G --group-add video --cap-add=SYS_PTRACE --security-opt seccomp=unconfined -v $HOME/dockerx:/dockerx'`
 - `drun rocm/tensorflow:latest`
4. Set environment variable "HIP_VISIBLE_DEVICES"
 - `export HIP_VISIBLE_DEVICES=0,1...N-1` where N is the number of multi-GPUs in your system.
 - For example, if we have two GPUs the variable is set as "export HIP_VISIBLE_DEVICES=0,1"

Running the Example

1. On the tutorial repo on your system run:
 - Clone the tutorial repo inside your docker container
 - `cd Chapter5/02_Keras_MultiGPU_ROCm/` in the tutorial repo
 - Based on the number of GPUs you have set in the environment variable `HIP_VISIBLE_DEVICES`, you will have to modify line number 36 accordingly.
 - For example; if you are using 2 GPUs it will look like this:
 - `model = multi_gpu_model(model, gpus=2, cpu_merge=True, cpu_relocation=False)`
 - Refer to the linked documentation for more info on this API call(https://www.tensorflow.org/api_docs/python/tf/keras/utils/multi_gpu_model)
2. Execute the script:
 - `python3 multi_gpu_mnist.py`

```

2020-07-22 22:48:00.319470: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library
libhip_hcc.so
2020-07-22 22:48:01.542555: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1579] Found device 0 with properties:
pciBusID: 0000:0d:00.0 name: Vega 20 ROCm AMD GPU ISA: gfx906
coreClock: 1.801GHz coreCount: 60 deviceMemorySize: 15.98GiB deviceMemoryBandwidth: -1B/s
2020-07-22 22:48:01.542620: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1579] Found device 1 with properties:
pciBusID: 0000:43:00.0 name: Vega 20 ROCm AMD GPU ISA: gfx906
coreClock: 1.801GHz coreCount: 60 deviceMemorySize: 15.98GiB deviceMemoryBandwidth: -1B/s
2020-07-22 22:48:01.947296: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library
librocblas.so
2020-07-22 22:48:01.948365: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library
libMIOpen.so
2020-07-22 22:48:02.410427: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library
librocfft.so
2020-07-22 22:48:02.412402: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library
librocrand.so
2020-07-22 22:48:02.412554: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1703] Adding visible gpu devices: 0, 1

```

Figure 1: Output of the Keras Model for training on the MNIST dataset using a 2-GPU system