

HIP Training Day 1 Exercises

1. Write a “Hello World” HIP program that uses printf from a GPU kernel. Compile and run this on the AMD Accelerator Cloud (AAC).
2. Write a HIP program that adds two vectors together on the GPU. You’ll need to
 - a. Create the vectors and assign them data on the host
 - b. Copy them to the device
 - c. Launch the GPU kernel
 - d. Copy the result back to the host
 - e. Check that your result is correct
3. Extend your solution to problem 2 by adding “hipEvents” to time the kernel.
 - a. Create a start and stop hipEvent
 - b. Record the start event before the kernel launch
 - c. Record the stop event after the kernel launch
 - d. Use hipEventSynchronize
 - e. Then use hipEventElapsedTime to compute the time between the start and stop events.
4. Choose one more of the CUDA samples in the Nvidia cloud and manually convert them to HIP. Compile and run them on the Nvidia cloud. Then, using `scp`, transfer them to the AAC by first copying to your local machine and then scp’ing to the AAC. Some suggestions include 0_Simple/<simplePrintf,vectorAdd,matrixMul,asyncAPI>, 1_Uilities/bandwidthTest
 - a. The CUDA samples are located in /usr/local/cuda/samples
 - b. You’ll need to copy the source file(s) to your local directory.
 - c. To compile on the Nvidia cloud, you can manually call `nvcc` or you can use `hipcc`. To use hipcc you’ll need to set `HIP_PLATFORM=nvidia` in your bash environment. In either the `nvcc` case or the `hipcc` case, you’ll need to include `-I/usr/local/cuda/samples/common/inc` on your compilation line
 - d. On the ACC, you’ll want to compile on the node you’ve been allocated so that `hipcc` will choose the correct GPU architecture.