EE 451: Parallel and Distributed Computation

PA5a — Spring 2021

Due date: Monday 29th March 2021 11:59 PM

1. **Examples**

   Copy example files to your home directory.

   1. Login to HPC
   2. Copy

      ```
      cp -r /project/xuehaiqi_652/cuda .
      ```

   The `hello.cu` contains the CUDA implementation of HelloWorld.

   1. Login to HPC
   2. Setup MPI toolchain:

      ```
      module purge
      module load gcc/8.3.0 cuda/10.1.243
      ```

   3. Compile

      ```
      nvcc -O3 hello.cu
      ```

   4. Run

      ```
      srun -n1 --gres=gpu:1 --mem=16G -t1 ./a.out
      ```

      The option `-t` specifies the limit of run time. Setting it as a small number will get your program scheduled earlier. The option `--mem` specifies the minimum memory requirement. Setting it as a small number will get your program scheduled earlier. However, you need it when you run `sumArraysOnGPU` and `sumMatrixOnGPU`. For more information on `srun` options, you can use `man srun` to find out.

   5. Profile (optional)

      ```
      srun -n1 --gres=gpu:p100:1 --partition=debug nvprof ./a.out
      ```

      If you want to do your final project with CUDA, you can try profiling. However, this is enabled only on P100 GPUs.

   6. Allocate a machine

      ```
      salloc -n1 --gres=gpu:1 --mem=16G -t10
      // After the allocation, you will log on the machine and have
         10 minutes to perform multiple operations
      ./a.out
      // edit, compile, and run again without waiting for a new
         allocation
      ./a.out
      ./a.out
      ```

If you want to do your final project with CUDA, you can try profiling. However, this is enabled only on P100 GPUs.

2. (100 points) 1. Remove the cudaDeviceReset function in hello.cu, then compile and run it.

   2. Replace the function cudaDeviceReset in hello.cu with cudaDeviceSynchronize, then compile and run it.

   3. In sumArraysOnGPU-timer.cu, set the block.x = 1023. Recompile and run it. Compare the result with the execution configuration of block.x = 1024. Try to explain the difference and the reason.

   4. In sumArraysOnGPU-timer.cu, let block.x = 256. Write a new kernel function that handles three elements. Compare the results with other grid configurations.

   5. In sumMatrixOnGPU.cu, consider 2D grid 1D block. Write a new kernel function that handle three elements. Find the best grid configuration.

   Submission Instructions: Submit your code, screenshots, and a performance report as described above.