|  |  |
| --- | --- |
| **Name:** | *Lakshya Goyal* |
| **NetID:** | *lgoyal3* |
| **Section:** | *AB* |

**ECE 408/CS483 Milestone 2 Report**

|  |
| --- |
| 1. Show output of rai running Mini-DNN on the basic GPU convolution implementation for batch size of 1k images. This can either be a screen capture or a text copy of the running output. Please do not show the build output. (The running output should be everything including and after the line "*Loading fashion-mnist data...Done*"). |
|  |
| 1. For the basic GPU implementation, list Op Times, whole program execution time, and accuracy for batch size of 100, 1k, and 10k images. |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Batch Size | Op Time 1 | Op Time 2 | Total Execution Time | Accuracy | | 100 | *0.249 ms* | *9.03 ms* | *1.181 s* | *86.00 %* | | 1000 | *2.36 ms* | *15.13 ms* | *9.877 s* | *88.60 %* | | 10000 | *23.27 ms* | *151.08 ms* | *1 min 38.494 s* | *87.14 %* | |
| 1. List all the kernels that collectively consumed more than 90% of the kernel time and what percentage of the kernel time each kernel did consume (start with the kernel that consumed the most time, then list the next kernel, until you reach 90% or more). |
| *Conv\_forward\_kernel 🡪 100 % 🡪 169.8 ms* |
| 1. List all the CUDA API calls that collectively consumed more than 90% of the API time and what percentage of the API time each call did consume (start with the API call that consumed the most time, then list the next call, until you reach 90% or more). |
| *cudaMemcpy 🡪 73.7 %*  cudaMalloc 🡪 13.0 %  cudaDeviceSynchronize 🡪 11.4 % |
| 1. Explain the difference between kernels and CUDA API calls. Please give an example in your explanation for both. |
| *The CUDA API calls are used by the CPU configure the kernel by either giving or getting data from the GPU. These execute on the CPU in order to do something to the GPU. Kernels run entirely on the GPU and so aren’t affected by any other CPU code.*  *Conv\_forward\_kernel is a kernel that doesn’t have any execution on the CPU, just the GPU. Whereas the cudaMemcpy() API function is called from the CPU to copy memory to the GPU.* |
| 1. Show a screenshot of the GPU SOL utilization |
| *<nsight output here>* |