CSC8014 Topics: GPU Programming

Homework #6

1. What is the main difference between `cudaMemcpy()` and `cudaMemcpyAsync()`?

   Ans:

   cudaMemcpy() is synchronous with host, the call will not return until memory copy is complete, it works on both pageable and page-locked host memory.

   cudaMemcpyAsync() is asynchronous with respect to host, it simply places a request of memory copy into the stream specified by the argument *stream,* so the call may return before the copy is complete. It works only on page-locked host memory and return an error if a pointer to pageable memory is passed as input. The copy is optionally associated with stream and may overlap with operations in other streams.

2. Is the following fragment of CUDA code correct?

```
cudaStream_t    stream;
int *a, *b, *c;
int *dev_a, *dev_b, *dev_c;

// initialize the stream
cudaStreamCreate( &stream );

// allocate the memory on the GPU
cudaMalloc( (void**)&dev_a, N * sizeof(int) );
cudaMalloc( (void**)&dev_b, N * sizeof(int) );
cudaMalloc( (void**)&dev_c, N * sizeof(int) );

// allocate host locked memory, used to stream
a = (int*)malloc( N * sizeof(int) );
b = (int*)malloc( N * sizeof(int) );
c = (int*)malloc( N * sizeof(int) );

for (int i=0; i<FULL_DATA_SIZE; i++) {
    a[i] = rand();
    b[i] = rand();
}

cudaMemcpyAsync( dev_a, a, N * sizeof(int),
                 cudaMemcpyHostToDevice, stream );
```

```
cudaMemcpyAsync( dev_b, b, N * sizeof(int),
                 cudaMemcpyHostToDevice, stream );

kernel<<<N/256,256,0,stream>>>( dev_a, dev_b, dev_c );

cudaMemcpyAsync( c, dev_c, N * sizeof(int),
                 cudaMemcpyDeviceToHost, stream );
```

Ans:

Since it needs to allocate page-locked host memory for stream, command cudaHostAlloc() should be used, the following code

```
// allocate host locked memory, used to stream
 a = (int*)malloc( N * sizeof(int) );
 b = (int*)malloc( N * sizeof(int) );
 c = (int*)malloc( N * sizeof(int) );
```

should be correct as:

```
// allocate host page-locked memory, used to stream
cudaHostAlloc((void **)&a, N * sizeof(int),cudaHostAllocDefault
);
cudaHostAlloc((void **)&b, N * sizeof(int), cudaHostAllocDefault
);
cudaHostAlloc((void **)&c, N * sizeof(int), cudaHostAllocDefault
);
```

3. What's the purpose of CUDA function `cudaStreamSynchronize( stream)`?

   Ans: in the main() program, when the *for()* loop terminates, there is no guarantee that all of tasks in the queue of stream are finished, there could still have quite a bit of work queued up for the GPU to finish, if we want to guarantee that the GPU is done with its computations and memory copies, we need to synchronize it with the host. That is, we want to ask host to sit around and wait for the GPU to finish before proceeding to next operation. The function cudaStreamSynchronize(stream) to accomplish the synchronize of GPU with host, argument *stream* is used to specify the stream that we want to wait for.

4. In the CUDA by Example, both `basic_double_stream.cu` and `basic_double_stream_correct.cu` define two streams (stream0 and stream1). Compile and run both programs. Observe the execution times for both programs. Do you see the differences? Explain why?