Memory Migration and Page Faults in CUDA

Memory shared between the host and the devices is called **Unified Memory**. After unified memory is allocated (by using *cudaMallocManaged* by example) it may not be mapped (or „known") to the CPU or the GPU. Therefore, when it is first mentioned, a **page fault** occurs which causes the requested memory to be migrated to the host or the device that wants to use it. After that, whenever memory that is a resident of a host or any device, it will automatically be migrated (page fault and on-demand memory migration occurs).

Because on-demand migration transfers data in small blocks it could be very inefficient when there is a lot of memory to be migrated. Because of that CUDA provides a feature called **asynchronous memory prefetching**. This technique allows programmers to migrate memory between devices and host. The entire process takes place in the background and the data is transfered in larger blocks of memory which reduces the time needed to finish this process. It can be achieved by using *cudaMemPrefetchAsync* function.

Below are the results generated with Nsight Systems after running a vector addition program. **1.1** shows a version with page fault memory migration and **1.2** shows a version using asynchronous memory prefetching:

1.1

```
Generating CUDA Kernel Statistics...
CUDA Kernel Statistics (nanoseconds)

Time(%)      Total Time   Instances         Average         Minimum         Maximum  Name
-------  --------------  ----------  --------------  --------------  --------------
-----------------------------------------------------------------------------------------------------------------------
-
  100.0       105364676           1     105364676.0       105364676       105364676  addVectorsInto(float*, float*,
float*, int)

Generating CUDA Memory Operation Statistics...
CUDA Memory Operation Statistics (nanoseconds)

Time(%)      Total Time  Operations         Average         Minimum         Maximum  Name
-------  --------------  ----------  --------------  --------------  --------------
-------------------------------------------------------------------------------
   85.7        68228825       11689          5837.0            2911           97792  [CUDA Unified Memory memcpy Ht
oD]
   14.3        11416288         768         14865.0            1887           93888  [CUDA Unified Memory memcpy Dt
oH]
```

1.2

```
Generating CUDA Kernel Statistics...
CUDA Kernel Statistics (nanoseconds)

Time(%)      Total Time   Instances         Average         Minimum         Maximum  Name
-------  --------------  ----------  --------------  --------------  --------------
-----------------------------------------------------------------------------------------------------------------------
-
  100.0          510530           1        510530.0          510530          510530  addVectorsInto(float*, float*,
float*, int)

Generating CUDA Memory Operation Statistics...
CUDA Memory Operation Statistics (nanoseconds)

Time(%)      Total Time  Operations         Average         Minimum         Maximum  Name
-------  --------------  ----------  --------------  --------------  --------------
-------------------------------------------------------------------------------
   77.8        37302144         192        194282.0          193216          195872  [CUDA Unified Memory memcpy Ht
oD]
   22.2        10664832          64        166638.0          166400          167328  [CUDA Unified Memory memcpy Dt
oH]
```

Observations: using asynchronous memory prefetching accelerated Kernel responsible for vector addition very significantly (around 200 times). There is also a significant decrease in number of memory migration operations performed (because they were done on bigger memory blocks).

Michał Orlewski