

Assignment I: GPU programming environment

Franz Kaschner

31.10.2022

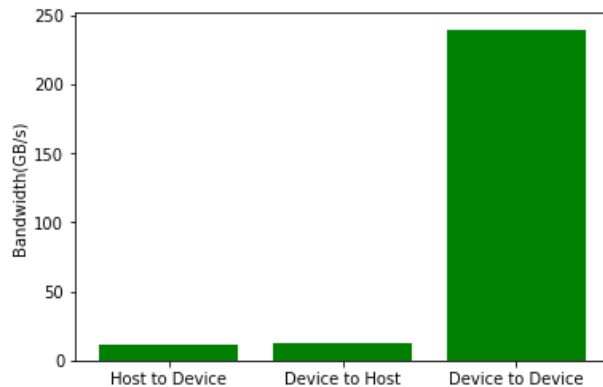
Tutorial: Using CUDA in Google Colab

Question: What GPU models did you get in your test?

Tesla T4

Exercise 2 - Bandwidth Test GPU-CPU on Google Colab

Bar plot in regular mode (transfer size 32000000 bytes):



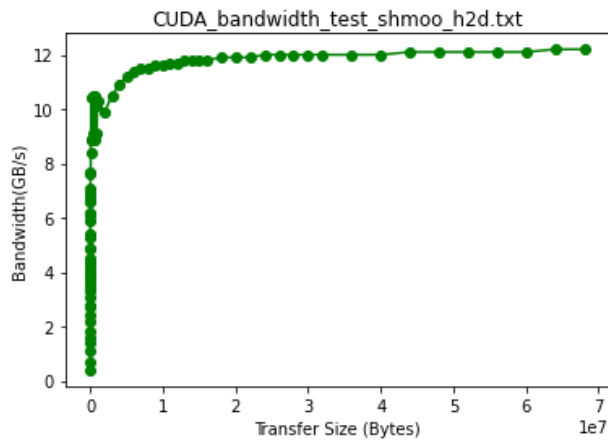
Explanation:

As it can be seen in the bar plot, the bandwidth of the device-to-device transfer is much higher than the bandwidth of the host-to-device and device-to-host transfers (approximately factor 20). The host-to-device and device-to-host transfers have a similar bandwidth. This is because the host-to-device and device-to-host communication uses PCIe which is much slower than the connection of the device to its own memory.

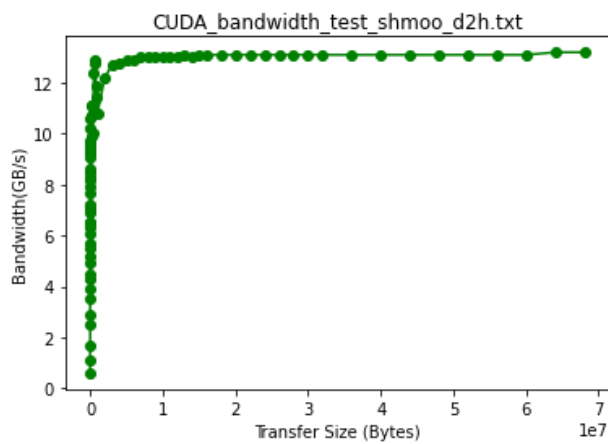
Host = CPU, device = GPU

Line plots in "shmoo" mode (increasing transfer sizes):

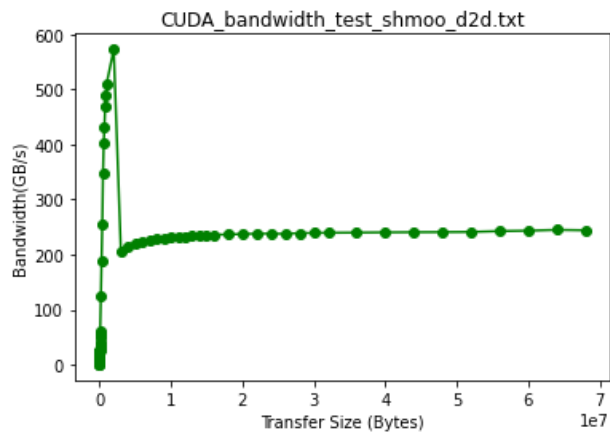
- Host to device (h2d):



- Device to host (d2h):



- Device to device (d2d):



Explanation:

As it can be seen in the line plots, for all three types of transfers, first, the bandwidth increases approximately linear and then at some point (e.g., at 3MB for the device-to-device case) the bandwidth drops. For the host-to-device and device-to-host transfers the drop is rather small and the bandwidth grows afterwards still to a higher level than before. At some point, the bandwidth saturates.

For the device-to-device transfer the drop is much bigger and the bandwidth does not reach the same speed afterwards as before. It saturates at a much lower level.

The explanation for the approx. linear growth at the beginning is probably that the “transfer time is the sum of a fixed overhead plus a variable portion growing linearly with the number of bytes transferred” ([source](#)). The fixed overhead is there because for smaller payloads the overhead due to the packet headers (of for example PCIe) is more significant. Furthermore, the transfer has to be initialized and that always takes the same amount of time independent of the transfer size. The saturation occurs because at some point the overhead converges to a minimum possible percentage (e.g., dependent on the packet size and the size of the header).

However, the reason of the drop is more complicated. One idea would be that with an increasing transfer size at some point a second packet is required which only carries a small payload. Hence the overhead due to the header would be large. However, if that would be the reason, you would get a periodic behavior where you reach the same speed after the drop as before.

A second idea could be that the DRAM memory of the GPU gets full with an increasing amount of data and that first data from the GPU has to be sent to the CPU before new data can be received. However, this argument is also not completely logical as the transfer size at the drops is much lower than the size of the DRAM memory of modern GPUs (typically several GBs).

The last possible reason could be that with a smaller transfer size the data still fully fits into the cache of the GPU. And with a higher transfer size the DRAM memory or a lower level cache has to be queried which takes more time which results in the drop.

Output of bandwidth test: ./bandwidthTest/bandwidthTest

```
[CUDA Bandwidth Test] - Starting...  
Running on...
```

```
Device 0: Tesla T4  
Quick Mode
```

```
Host to Device Bandwidth, 1 Device(s)  
PINNED Memory Transfers  
Transfer Size (Bytes)  Bandwidth(GB/s)  
32000000              11.9
```

```
Device to Host Bandwidth, 1 Device(s)  
PINNED Memory Transfers  
Transfer Size (Bytes)  Bandwidth(GB/s)  
32000000              12.9
```

```
Device to Device Bandwidth, 1 Device(s)  
PINNED Memory Transfers  
Transfer Size (Bytes)  Bandwidth(GB/s)  
32000000              239.6
```

```
Result = PASS
```

```
NOTE: The CUDA Samples are not meant for performance measurements. Results may vary when GPU Boost is enabled.
```

Output of bandwidth test in "shmoo" mode: ./bandwidthTest/bandwidthTest --mode=shmoo

```
[CUDA Bandwidth Test] - Starting...  
Running on...
```

```
Device 0: Tesla T4  
Shmoo Mode
```

```
.....  
Host to Device Bandwidth, 1 Device(s)  
PINNED Memory Transfers  
Transfer Size (Bytes)  Bandwidth(GB/s)  
1000                  0.5  
2000                  0.9  
3000                  1.3  
4000                  1.7  
5000                  2.0  
6000                  2.1  
7000                  2.4  
8000                  2.9  
9000                  3.0  
10000                 3.3  
11000                 3.5  
12000                 3.8  
13000                 4.0  
14000                 4.2  
15000                 4.4  
16000                 4.4  
17000                 4.5  
18000                 4.6  
19000                 4.9  
20000                 5.0  
22000                 4.7  
24000                 5.3  
26000                 5.3  
28000                 5.3  
30000                 5.6  
32000                 5.6  
34000                 6.2  
36000                 6.1  
38000                 6.6  
40000                 6.4  
42000                 6.5  
44000                 6.8  
46000                 6.9  
48000                 7.1  
50000                 7.2  
60000                 7.8
```

70000	1.0	
80000	8.6	
90000	8.9	
100000		9.1
200000		7.9
300000		10.2
400000		8.6
500000		9.5
600000		8.6
700000		9.6
800000		8.8
900000		9.6
1000000		9.2
2000000		9.5
3000000		9.3
4000000		9.9
5000000		10.0
6000000		10.4
7000000		10.5
8000000		10.6
9000000		10.8
10000000		10.9
11000000		11.1
12000000		11.1
13000000		11.1
14000000		11.4
15000000		11.4
16000000		11.4
18000000		11.5
20000000		11.5
22000000		11.5
24000000		11.6
26000000		11.7
28000000		11.7
30000000		11.7
32000000		11.8
36000000		11.9
40000000		11.9
44000000		11.9
48000000		11.9
52000000		11.9
56000000		11.9
60000000		12.0
64000000		12.0
68000000		12.0

.....

Device to Host Bandwidth, 1 Device(s)

PINNED Memory Transfers

Transfer Size (Bytes)	Bandwidth(GB/s)
1000	0.5
2000	1.1
3000	1.6
4000	2.3
5000	3.0
6000	3.0
7000	3.7
8000	4.3
9000	4.3
10000	5.0
11000	5.3
12000	5.6
13000	5.8
14000	6.0
15000	6.3
16000	6.5
17000	6.7
18000	6.6
19000	7.1
20000	6.9
22000	7.4
24000	7.8
26000	8.1
28000	8.3
30000	8.5
32000	8.7
34000	8.7

36000	9.1	
38000	9.2	
40000	9.4	
42000	9.5	
44000	9.5	
46000	9.7	
48000	9.8	
50000	9.9	
60000	10.1	
70000	10.6	
80000	10.8	
90000	11.1	
100000		11.2
200000		10.1
300000		9.0
400000		10.2
500000		10.5
600000		12.4
700000		12.2
800000		11.4
900000		11.3
1000000		10.6
2000000		11.1
3000000		11.9
4000000		12.0
5000000		12.2
6000000		12.5
7000000		12.6
8000000		12.7
9000000		12.8
10000000		12.8
11000000		12.9
12000000		12.9
13000000		12.9
14000000		12.9
15000000		12.9
16000000		13.0
18000000		13.0
20000000		13.0
22000000		13.1
24000000		13.1
26000000		13.1
28000000		13.1
30000000		13.1
32000000		13.1
36000000		13.1
40000000		13.1
44000000		13.1
48000000		13.1
52000000		13.1
56000000		13.1
60000000		13.1
64000000		13.1
68000000		13.2

.....

Device to Device Bandwidth, 1 Device(s)

PINNED Memory Transfers

Transfer Size (Bytes)	Bandwidth(GB/s)
1000	0.5
2000	1.1
3000	1.7
4000	1.9
5000	2.9
6000	3.5
7000	4.1
8000	4.8
9000	5.3
10000	6.0
11000	5.0
12000	6.7
13000	7.7
14000	7.5
15000	9.1
16000	8.8
17000	9.5
18000	9.7

19000	10.6	
20000	12.1	
22000	12.7	
24000	14.4	
26000	15.5	
28000	16.9	
30000	18.5	
32000	17.4	
34000	18.1	
36000	20.4	
38000	22.2	
40000	24.1	
42000	24.8	
44000	24.1	
46000	26.6	
48000	28.2	
50000	30.1	
60000	37.6	
70000	40.8	
80000	47.2	
90000	55.3	
100000		58.5
200000		120.8
300000		183.6
400000		231.5
500000		308.1
600000		336.0
700000		431.9
800000		466.8
900000		487.6
1000000		508.0
2000000		548.8
3000000		207.5
4000000		214.4
5000000		220.3
6000000		223.5
7000000		226.3
8000000		228.5
9000000		229.4
10000000		231.0
11000000		232.0
12000000		232.7
13000000		234.1
14000000		234.8
15000000		235.4
16000000		235.7
18000000		236.5
20000000		237.2
22000000		237.7
24000000		237.9
26000000		238.2
28000000		238.7
30000000		239.0
32000000		239.5
36000000		239.8
40000000		240.2
44000000		240.5
48000000		240.6
52000000		240.9
56000000		241.0
60000000		241.1
64000000		242.3
68000000		241.4

Result = PASS

NOTE: The CUDA Samples are not meant for performance measurements. Results may vary when GPU Boost is enabled.