

Paměťové transfery a jednoduché kernely

PCG – Paralelní výpočty na GPU

Cvičení 1, 2022/2023

Jirka Jaroš

Vysoké učení technické v Brně, Fakulta informačních technologií
Božetěchova 1/2, 612 66 Brno - Královo Pole
jarosjir@fit.vutbr.cz



UKÁZKY CUDA UTILS

- **Připojte se na Karolínu**

```
ssh karolina
```

- **Vaše PC: Připojte si disk z Karolíny a nakopírujte tam obsah 1. cvičení**

```
mkdir /tmp/karolina  
sshfs karolina: /tmp/karolina
```

- **Nastartujte job (pozor, máme jen 8 uzlů). Kompilovat lze i na loginu**

- středa – 12.10.

```
qsub -q R1519443 -A DD-22-68 -I -X -l walltime=1:00:0 -l select=1
```

- pátek – 14.10.

```
qsub -q R1519444 -A DD-22-68 -I -X -l walltime=1:0:0 -l select=1
```

- Kdykoliv jindy

```
qsub -q qgpu -A DD-22-68 -I -X -l walltime=1:0:0 -l select=1
```

- **Natáhněte moduly**

```
ml CUDA/11.1.1-GCC-10.2.0 Qt5/5.14.2-GCCcore-10.2.0
```

- **Export cest ke CUDA**

```
export PATH=$PATH:/usr/local/share/cuda-11.4/bin
```

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/share/cuda-  
11.4/bin:/usr/local/share/cuda-11.4/targets/x86_64-  
linux/lib:/usr/local/share/cuda-11.4/extras/CUPTI/lib64
```

- **Pro export cest použijte script**

```
cd Utilities  
chmod +x path_export.sh  
eval `./path_export.sh`
```

1. Přeložte knihovnu libwb – bude se hodit pro další ukázky

```
cd libwb  
make -j  
make libwb.a
```

1. Zjistěte co za GPU máte

```
nvidia-smi
```

1. Přeložte deviceQuery (CVT)

```
cd Utilities/deviceQuery  
make -j
```

2. Přeložte bandwidthTest (nelze v CVT)

```
cd Utilities/deviceQuery  
make -j
```

- Spustíte program deviceQuery

```
./deviceQuery
```

- Výstup

```
./deviceQuery Starting...

CUDA Device Query (Runtime API) version (CUDA static linking)

Detected 1 CUDA Capable device(s)

Device 0: "Tesla K20m"
  CUDA Driver Version / Runtime Version      10.2 / 10.2
  CUDA Capability Major/Minor version number: 3.5
  Total amount of global memory:              4744 MBytes (4974313472
bytes)
  (13) Multiprocessors, (192) CUDA Cores/MP: 2496 CUDA Cores
  GPU Max Clock rate:                        706 MHz (0.71 GHz)
  Memory Clock rate:                          2600 Mhz
  Memory Bus Width:                           320-bit
Result = PASS
```

Jaké jsou
parametry
GPU?

- Spustíte program `bandwidthTest`

```
./bandwidthTest --mode=shmoo --memory=pinned  
./bandwidthTest --mode=shmoo --memory=pageable
```

- Výstup

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

```
Running on...
```

```
Device 0: Tesla K20m
```

```
Shmoo Mode
```

```
.....  
Host to Device Bandwidth, 1 Device(s)
```

```
PINNED Memory Transfers
```

Transfer Size (Bytes)	Bandwidth (MB/s)
1024	364.2
2048	784.7
3072	1144.5
4096	1474.8
5120	1869.5
6144	2132.1

Jakých
rychlostí
dosahujete?

ÚLOHY

- **Otevřete soubor template.cu**

- Doplňte kernel tak, aby vypsal ID vlákna a bloku

```
/**
 * Write a CUDA kernel and run it.
 */
void cudaHelloWorld(void)
{
    int bx = 0;
    int tx = 0;

    printf("Hello world. I'm blockIdx.x = %d, threadIdx.x = %d\n.", bx, tx);
} // end of cudaHelloWorld
```

- Zajistěte správné zavolání kernelu ve funkci main

- **Překlad a spuštění (CVT)**

```
make
make run
```

- **Překlad a spuštění (Karolíně – upravte ve scriptu číslo projektu)**

```
qsub run.pbs
```

Výstupem je
výpis ID
vlákna a
bloku

- **Otevřete soubor template.cu**
 - Doplněte kernel tak, aby sečetl dva vektory
 - Doplněte kód pro alokaci data na GPU
 - Doplněte kód pro kopii data na GPU
 - Spustěte kernel
 - Stáhněte data zpět na CPU
- **Překlad a spuštění pro různé velikosti problému**
 - Vygenerujte vstupní data

```
make
make gen
make run0
...
make run6
```

```
make run0
./ex2 -e VectorAdd/Dataset/0/output.raw -i
VectorAdd/Dataset/0/input0.raw,VectorAdd/Dataset/0/input1.raw -t vector
----- Example 2: CUDA vector addition -----
Number of elements: 16
==$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
{
  "timer":[],
  "logger":[],
  "solution_exists": true,
  "solution":{
    "correctq": true,
    "message": "Solution is correct."
  }
}
```

```
qsub run.pbs
```

- Vygenerujte vstupní data

```
make run0
./ex3 -e
ImageColorToGrayscale/Dataset/0/output.pbm -i
ImageColorToGrayscale/Dataset/0/input.ppm -t
image
- Example 3: Image conversion from RGB to gray-
Image size [256, 256]
==$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
{
"timer":[],
"logger":[],
"solution_exists": true,
"solution":{
"correctq": true,
"message": "Solution is correct."
}
}
```

11

- Spuštění grafického rozhraní:

```
nsys-ui
```

- Spouštění v příkazové řádce:

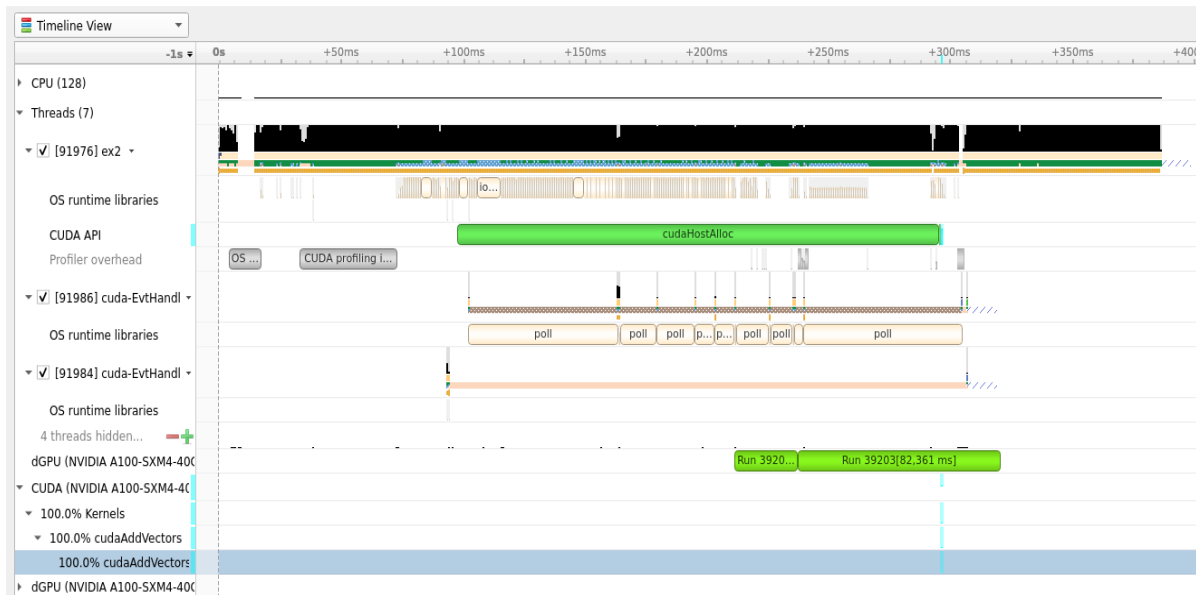
```
nsys
```

- Vytvořte nový projekt

- Nastavte hodnotu v poli: *Command line with argument*

```
/home/.../ex3 -e ImageColorToGrayscale/Dataset/5/output.pbm -i ImageColorToGrayscale/Dataset/5/input.ppm
```

- Označte: *Collect CUDA trace*



- Spuštění grafického rozhraní:

```
ncu-ui
```

- Spouštění v příkazové řádce:

```
ncu
```

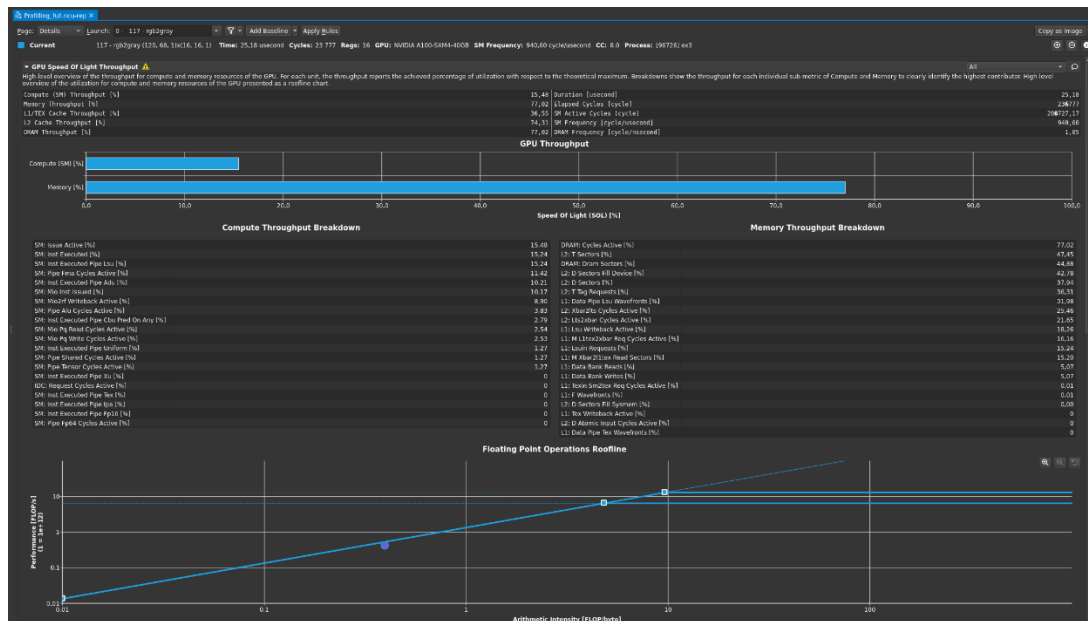
- Vytvořte nový projekt
 - Zvolte spustitelný soubor:

```
/home/.../ex3
```

- Argumenty programu:

```
-e ImageColorToGrayscale/Dataset/5/output.pbm -i ImageColorToGrayscale/Dataset/5/input.ppm
```

- Nastavte, které metriky vás zajímají



Vzorové řešení zveřejníme v pátek