

Práce s pamětí textur a pamětí konstant

PCG – Paralelní výpočty na GPU

Cvičení 3, 2022/2023

Jirka Jaroš

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



- **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`
```

PŘIPOJENÍ NA KAROLÍNU

- **Připojte se na Karolínu**

```
ssh karolina
```

- **Vaše PC: Připojte si disk z Barbory a nakopírujte tam obsah 3. cvičení**

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

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

- středa – 02.11.

```
qsub -q R1572268 -A DD-22-68 -I -X -l walltime=1:00:0 -l select=1
```

- pátek – 04.11.

```
qsub -q R1572271 -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**

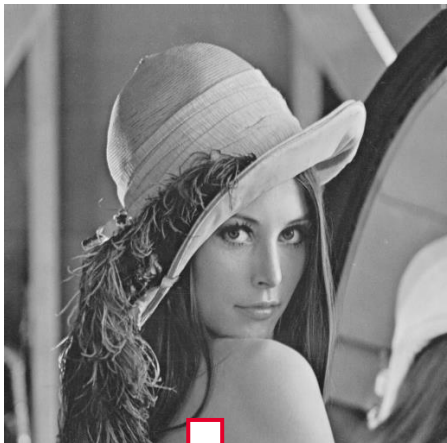
```
m1 CUDA/11.1.1-GCC-10.2.0 Qt5/5.14.2-GCCcore-10.2.0
```

1. Zkopírujte zdrojové kódy na Karolínu

2. Kihovna libwb dnes není potřeba

**3. Implementaci a testování funkčnosti můžete dělat v CVT,
benchmarkování a profilování na Karolíně pomocí skriptu run.pbs**

ROTACE OBRAZU POMOCÍ TEXTUR



Výpočet

```
// x, y - coordinates in the rotated image
// u, v - coordinates in the original image

float u = x / float(width);
float v = y / float(height);

// Move coordinates to the middle of the image
u -= 0.5f;
v -= 0.5f;

// Calculate rotated coordinates
const float tu = u * cosf(theta) - v * sinf(theta) + 0.5f;
const float tv = v * cosf(theta) + u * sinf(theta) + 0.5f;

outputImage[y * width + x] = tex2D<float>(texImage, tu, tv);
```

- **Otevřete soubor template.cu**

- Definujte 2D texturu nad datovým typem `float`
- Doplňte do kernelu načítání z textury
- Definujte formát deskriptoru kanálu

```
cudaChannelFormatDesc
```

- Nastavte vhodné parametry textury
 - mód interpolace
 - filtrování
 - přístup přes normalizované koordináty
 - [Cuda Texture documentation](#)

- Vytvořte objekt textury na globální pole

```
cudaCreateTextureObject
```

- Po ukončení výpočtu odstraňte objekt textury

```
cudaDestroyTextureObject
```

Překlad a spuštění CVT (bez profil)

```
make  
make run{1,2}  
make prof{1,2}
```

Karolína (profil)

```
qsub run.pbs
```

Upravte soubor:

```
make  
make prof{1,2}
```


- **Otevřete soubor `template.cu`**

- Definujte 2D texturu nad datovým typem `float`
- Definujte formát deskriptoru kanálu
- Alokujte cuda array (`cudaMallocArray`)
- Nakopírujte zdrojový obrázek do cuda array

`cudaMemcpy2DToArray`

- Nastavte vhodné parametry textury
- Vytvořte objekt textury na globální pole

`cudaCreateTextureObject`

- Po ukončení výpočtu odstraňte objekt textury

`cudaDestroyTextureObject`

Překlad a spuštění

CVT (bez profil)

```
make
make run{1,2}
make prof{1,2}
```

Karolína (profil)

```
qsub run.pbs
```

Upravte soubor:

```
make
make prof{1,2}
```

- **Otevřete soubor `template.cu`**

- Definujte C++ strukturu `ConstantParams` udržující výšku, šířku a úhel rotace.
- Definujte objekt `cudaParams` v paměti konstant GPU.
- Upravte CUDA kernel tak, aby využíval paměť konstant
- Naplňte paměť konstant pomocí `cudaMemcpyToSymbol`

Překlad a spuštění

CVT (bez profil)

```
make  
make run{1,2}  
make prof{1,2}
```

Karolina (profil)

```
qsub run.pbs
```

Upravte soubor:

```
make  
make prof{1,2}
```

**Vzorové řešení zveřejníme v
pátek**