

CORSO HPC: ESERCITAZIONE HEAT TRANSFER

Alessandro Dal Palù¹

1. Dipartimento di Scienze Matematiche, Fisiche e Informatiche,
Università di Parma

- Le informazioni sono contenute nel libro CUDA by Example
- due matrici di appoggio: new/old (usate come doppio buffer: si legge da una per fare i conti, si scrive nell'altra il risultato)
- Utilizziamo 2 kernel:
- il primo copia i dati costanti (l'oggetto con la sua temperatura)
- il secondo calcola l'iterazione

SETUP

PREPARARE L'AMBIENTE

- Per poter generare gli eseguibili per una GPU, bisogna preparare l'ambiente
- Lanciare `> module load cuda`
- I files della lezione sono in
`/hpc/home/alessandro.dalpalu/gpu/heat`
- Lanciare

```
cd
mkdir gpuheat
cd gpuheat
cp -R /hpc/home/alessandro.dalpalu/gpu/heat/* .
```

SETUP

PREPARARE L'AMBIENTE

- Per compilare lanciare

```
nvcc -O2 heat_gpu.cu -o heat_gpu
```

PROVARE L'ESEMPIO

```
> cat heat_gpu.slurm

#!/bin/sh
# Richiedi un nodo con una gpu
#SBATCH --partition=gpu
#SBATCH --nodes=1
#SBATCH --mem=4G
#SBATCH --qos=gpu
#SBATCH --gres=gpu:p100:1
# Dichiaro che il job durera' al massimo 1 minuto
#SBATCH --time=0-00:01:00
# stampa il nome del nodo assegnato e argomenti
echo "#SLURM_JOB_NODELIST      : $SLURM_JOB_NODELIST"
echo "#CUDA_VISIBLE_DEVICES    : $CUDA_VISIBLE_DEVICES"
# esegui il programma
module load cuda
./bin/heat_gpu
```

LANCIARE L'ESEMPIO

```
> sbatch slurm_launch_single  
> cat slurm-XXXXX.out | fgrep -v '#' > my_file.csv  
> python heat_plot.py my_file.csv
```

Viene generato un file png contenente la matrice disegnata

CONSEGNA 1

- Aggiungere un timer per misurare il tempo di calcolo delle iterazioni dei kernel
- Aggiungere in output il tempo preceduto dal simbolo # (per permettere a grep di generare il csv)
- Modificare l'inizializzazione della matrice Initialize() per creare la propria configurazione (i valori >0 vengono imposti nella simulazione)

CONSEGNA 2

- 1 usare parametro da riga di comando per definire la dimensione della matrice (non 256 statica) e mostrare andamento dei tempi con un grafico per $N=64,128,256,512,1024,2048$
- 2 introdurre uso memoria shared per leggere T_{old} e mostrare l'andamento dei tempi al variare della dimensione della matrice
- 3 cambio dimensione del blocco $BLOCK_SIZE_X$ (4,8,16,32) e mostrare l'andamento dei tempi al variare della dimensione della matrice

CONSEGNA 3

Opzionale

- Usare le texture per l'accesso ai dati