**Lập trình song song trên GPU**

**Hoàng Minh Thanh (18424062)**



**TH3 : Bộ nhớ trong CUDA**

C:\Users\tdqua_000\Dropbox\SS-Slides\DeCuong-CDIO\Template CDIO v4.2\Templates\Hinh anh\LogoTruong.png

Bộ môn Công nghệ phần mềm

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

Contents

[I. Quá trình cài đặt 3](#_Toc48744644)

[1) Hàm tính blurImgKernel1 3](#_Toc48744645)

[2) Hàm blurImgKernel2 sử dụng SMEM 5](#_Toc48744646)

[3) Hàm blurImgKernel3 sử dụng CMEM 7](#_Toc48744647)

[4)Chạy các kích thước khác nhau: 7](#_Toc48744648)

[5) Giải thích kết quả : 10](#_Toc48744649)

[1. Sử dụng SMEM sẽ nhanh hơn so với không dùng: 11](#_Toc48744650)

[2. CMEM lại chạy nhanh hơn so với không dùng 11](#_Toc48744651)

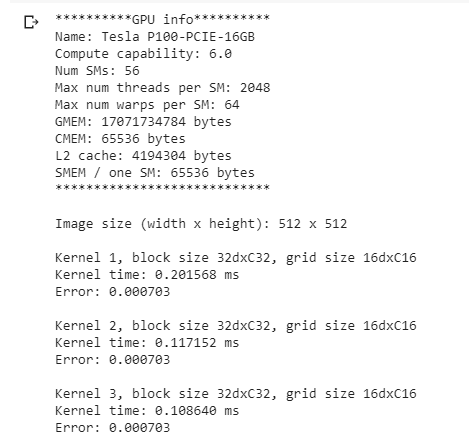
Vì máy tính cả nhân của em không có GPU nên bắt buộc em phải sử dụng Google Colab :

<https://colab.research.google.com/drive/1c1hbh_t62g-YkdI1uLWnlj7pi7qNU6hF#scrollTo=cu3t0-tTgI0E> (Thầy có thể vào link Online để xem luồng chạy dễ hơn báo cáo )

# Quá trình cài đặt

Chi tiết cài đặt trong file .cu và file Google Colab

Kết quả toàn bộ chương trình :



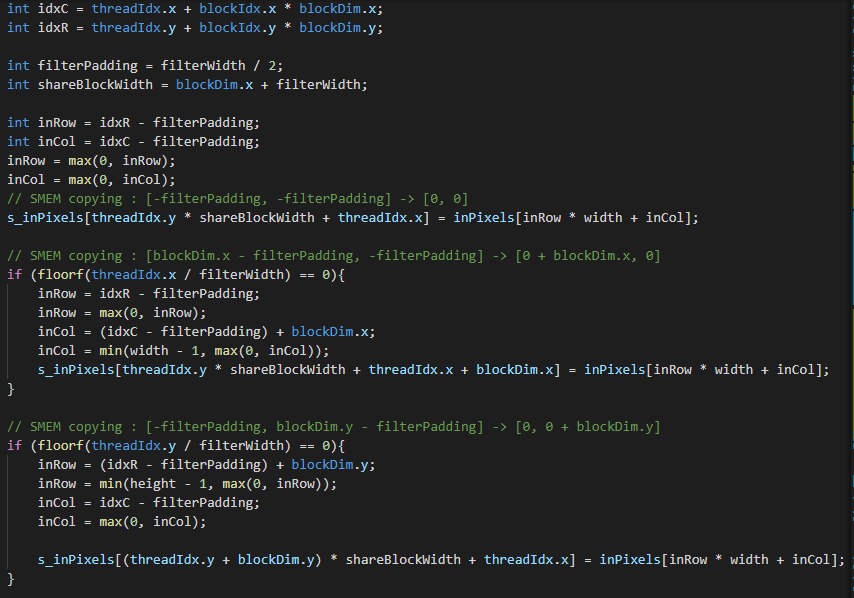
## Hàm tính blurImgKernel1

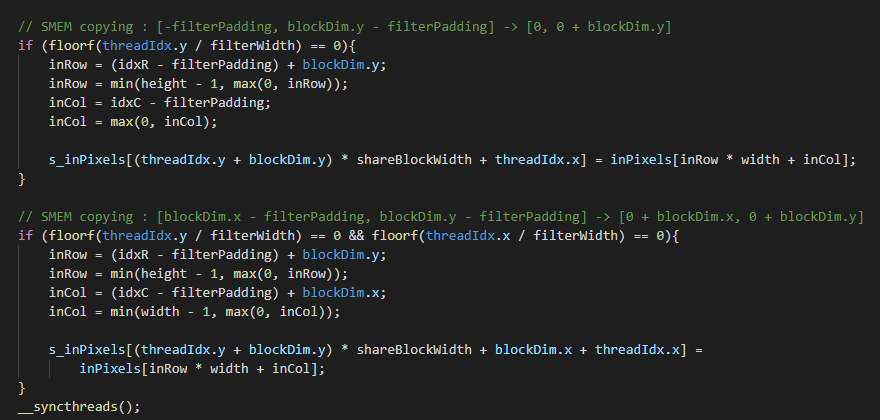
Kết quả cài đặt

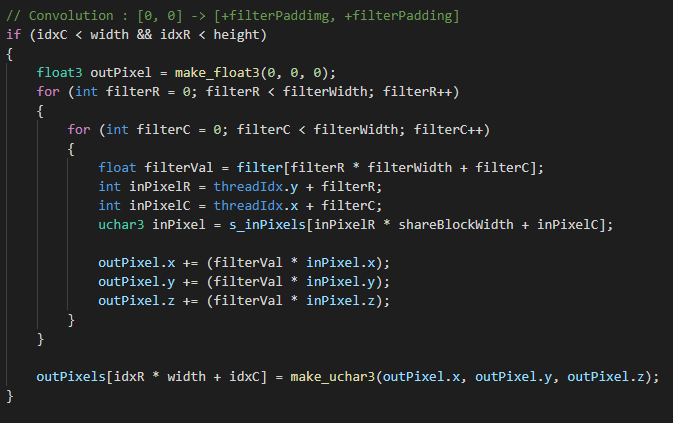
Chương trình cài đặt

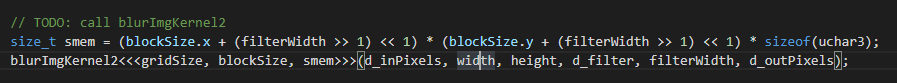


## Hàm blurImgKernel2 sử dụng SMEM



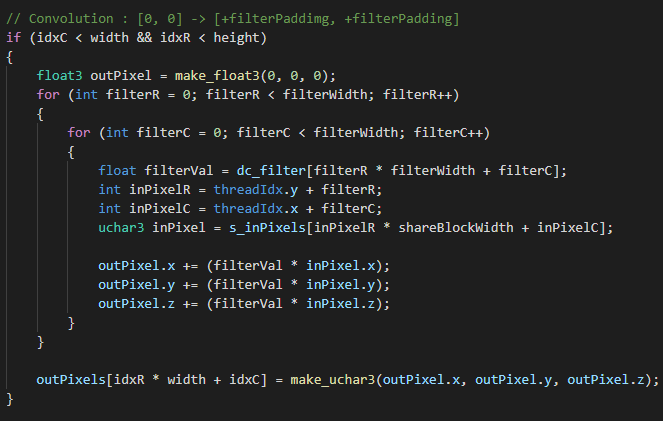




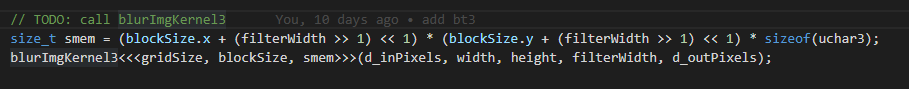


## 3) Hàm blurImgKernel3 sử dụng CMEM

Hàm copy CMEM

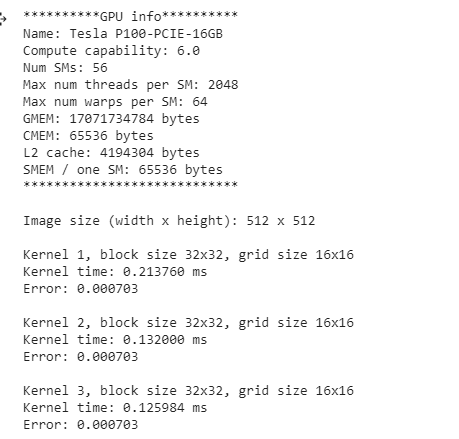


Hàm gọi blurImgKernel3

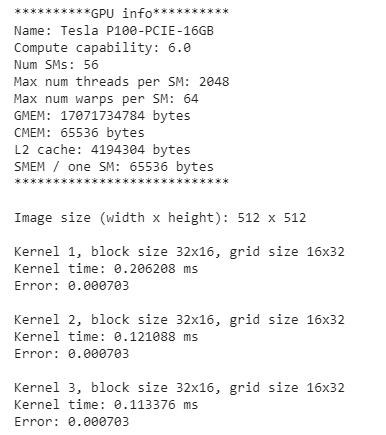


## 4)Chạy các kích thước khác nhau:

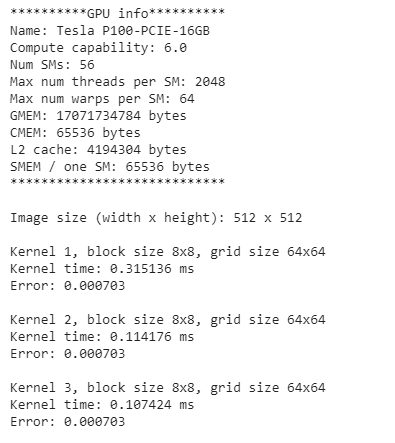
32 x 32



32 x 16



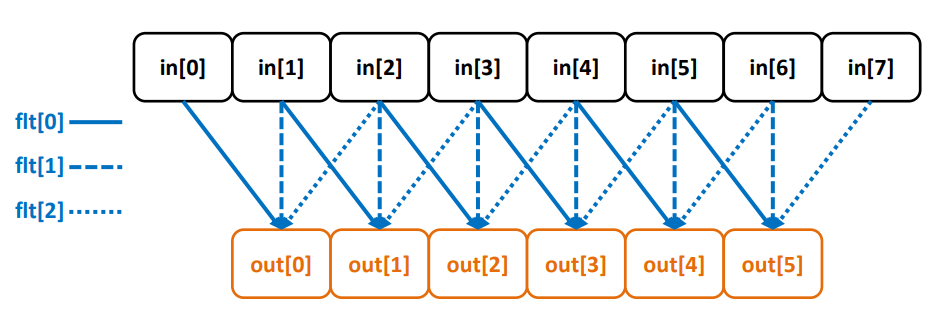
8x8



## 5) Giải thích kết quả :

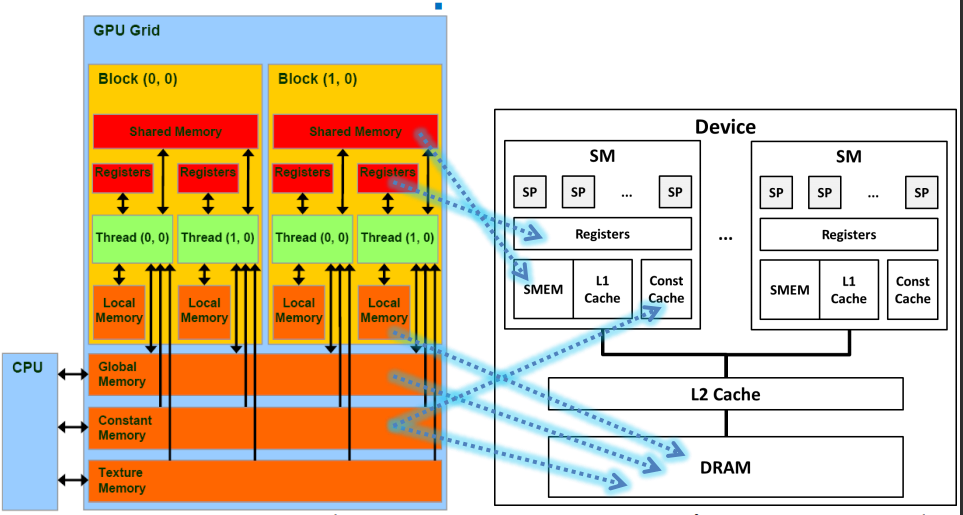
Giải thích tại sao kết quả lại như vậy (tại sao dùng SMEM lại chạy nhanh hơn so với không dùng, tại sao dùng).

### 1. Sử dụng SMEM sẽ nhanh hơn so với không dùng:



*Ví dụ với convolution 1 chiều*

Vì khi copy dữ liệu từ GMEM sang SMEM thì với mỗi lần tính convolution thì thay vì các thread phải lên GMEM (nơi có tốc độ truy xuất chậm) thì các thread sẽ chạy lên SMEM là bộ nhớ chia sẻ trong mỗi block thì sẽ có kết quả truy xuất nhanh hơn và không phải chạy đến GMEM để lấy dữ liệu nhiều lần.



### 2. CMEM lại chạy nhanh hơn so với không dùng

Vì CMEM là bộ nhớ hằng, ở trong bộ nhớ hằng sẽ không thay đổi giá trị khi đã gán là bộ nhớ được tối ưu để lưu các giá trị không thay đổi, và sẽ được lưu dưới một địa chỉ trong quá trình tính toán. Vì vậy nếu các thread trong cùng một wrap sẽ gọi đến cùng một địa chỉ, và chỉ tốn 1 lần đọc duy nhất. Sau đó địa chỉ đó sẽ được broadcast cho các thread trong một wrap. Nên từ đó tối ưu tốc độ tính toán. Nên có tốc độ chạy nhanh hơn so với không dùng.