Lập trình song song trên GPU
BT01: CUDA cơ bản (phần 1)

Nên nhớ mục tiêu chính ở đây là học, học một cách chân thật. Bạn có thể thảo luận ý tưởng với bạn khác, nhưng bài làm phải là của bạn, dựa trên sự hiểu thật sự của bạn. Nếu vi phạm thì sẽ bị 0 điểm cho toàn bộ môn học.

Đề bài

Viết chương trình chuyển ảnh RGB (ảnh màu) sang ảnh grayscale (ảnh xám) theo công thức:

giá-trị-grayscale = 0.299×giá-trị-red + 0.587×giá-trị-green + 0.114×giá-trị-blue

Mình có đính kèm:

- File ảnh đầu vào RGB "in.pnm" (trong Windows, bạn có thể xem file *.pnm bằng chương trình IrfanView).
- File khung chương trình "bt01_p1.cu". Chương trình sẽ:
  - Đọc file ảnh đầu vào RGB.
  - Chuyển ảnh đầu vào RGB sang ảnh grayscale vào bằng host (làm tuần tự).
  - Chuyển ảnh đầu vào RGB sang ảnh grayscale bằng device (làm song song). *Bạn sẽ phải code phần này, cụ thể là ở những chỗ mình để "// TODO". Lưu ý: bạn cần kiểm lỗi khi gọi các hàm CUDA API (dùng macro CHECK mà mình đã viết sẵn cho bạn) và khi gọi hàm kernel.*
  - So sánh ảnh kết quả của device với host để giúp bạn biết là đã cài đặt đúng hay chưa.
  - Ghi các ảnh kết quả xuống file.

Hướng dẫn về các câu lệnh

(Phần hướng dẫn dưới đây là cho Linux. Nếu bạn nào dùng GPU cá nhân trên Windows thì cũng tương tự, chỉ khác là: file chạy sau khi biên dịch có đuôi exe, và khi chạy file này thì dùng .\ thay vì ./)

- Biên dịch file "bt01_p1.cu": nvcc bt01_p1.cu -o bt01_p1
  Câu lệnh này sẽ biên dịch file "bt01_p1.cu" bằng trình biên dịch nvcc của NVIDIA, và xuất ra file chạy "bt01_p1" (nếu bạn muốn xuất ra file chạy có tên khác thì sau -o bạn thay bt01_p1 bằng tên mà bạn muốn; nếu bạn chỉ gõ là nvcc bt01_p1.cu thì sẽ xuất ra file chạy có tên mặc định là "a.out")
- Chạy file "bt01_p1" với file ảnh đầu vào là "in.pnm" và xuất ảnh kết quả ra file "out.pnm" (kết quả của host sẽ xuất ra file "out_host.pnm", còn device thì là "out_device.pnm"): ./bt01_p1 in.pnm out.pnm
  Chương trình sẽ in ra thời gian thực thi của host và của device, và giá trị khác biệt trung bình giữa ảnh kết quả của host và của device (được tính bằng cách: lấy các pixel tương ứng của 2 ảnh trừ cho nhau, lấy trị tuyệt đối, và cuối cùng tính trung bình hết lại). Như vậy, nếu giá trị khác biệt trung bình bằng 0 thì nghĩa là 2 ảnh giống hệt nhau. Nếu giá trị khác biệt trung bình có giá trị nhỏ (ví dụ, 0.xxx) thì chưa

chắc là sai, vì khi tính toán số thực thì giữa CPU và GPU có thể có sai biệt nhỏ; nhưng nếu giá trị khác biệt trung bình lớn hơn 0.xxx thì chắc là sai rồi.

Mặc định thì sẽ dùng block có kích thước 32×32; nếu bạn muốn chỉ định kích thước block thì truyền thêm vào câu lệnh 2 con số lần lượt ứng với kích thước theo chiều x và theo chiều y của block (ví dụ, ./bt01_p1 in.pnm out.pnm 32 16).

Nộp bài

Ở buổi sau, sẽ có phần 2. Khi đó, bạn sẽ nộp một lần cả phần 1 lẫn phần 2. Nhưng bạn nên hoàn thành phần 1 trong tuần này.