Lập trình song song trên GPU
HW2: Cách thực thi song song trong CUDA

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.

Trong môn học, để thống nhất, tất cả các bạn (cho dù máy bạn có GPU) đều phải dùng Google Colab để biên dịch và chạy code (khi chấm Thầy cũng sẽ dùng Colab để chấm). Với mỗi bài tập, bạn thường sẽ phải nộp:

1) **File code** (file .cu)
2) **File báo cáo** là file notebook (file .ipynb) của Colab (nếu bạn nào biết Jupyter Notebook thì bạn thấy Jupter Notebook và Colab khá tương tự nhau, nhưng 2 cái này hiện chưa tương thích 100% với nhau: file .ipynb viết bằng Jupyter Notebook có thể sẽ bị mất một số cell khi mở bằng Colab và ngược lại). File này sẽ chứa các kết quả chạy. Ngoài ra, một số bài tập có phần viết (ví dụ, yêu cầu bạn nhận xét về kết quả chạy), và bạn sẽ viết trong file notebook của Colab luôn. Colab có 2 loại cell: code cell và text cell. Ở code cell, bạn có thể chạy các câu lệnh giống như trên terminal của Linux bằng cách thêm dấu ! ở đầu. Ở text cell, bạn có thể soạn thảo văn bản theo cú pháp của Markdown (rất dễ học, bạn có thể xem ở đây); như vậy, bạn sẽ dùng text cell để làm phần viết trong các bài tập. Bạn có thể xem về cách thêm code/text cell và các thao tác cơ bản ở đây, mục "Cells" (đừng đi qua mục "Working with Python"). Một phím tắt ưa thích của mình khi làm với Colab là ctrl+shift+p để có thể search các câu lệnh của Colab (nếu câu lệnh có phím tắt thì bên cạnh kết quả search sẽ có phím tắt). File notebook trên Colab sẽ được lưu vào Google Drive của bạn; bạn cũng có thể download trực tiếp xuống bằng cách ấn ctrl+shift+p, rồi gõ "download .ipynb".

Đề bài

Câu 1 (4đ)

Áp dụng những hiểu biết về cách thực thi song song trong CUDA để tối ưu hóa chương trình thực hiện "reduction" (cụ thể: ta sẽ tính tổng của một mảng số nguyên).

Code (2đ)

Mình đã viết sẵn cho bạn khung chương trình trong file "HW2_P1.cu" đính kèm; bạn chỉ viết code ở những chỗ có từ "// TODO":

- Hàm kernel 1, 2, và 3 (bạn xem nội dung cụ thể của các hàm kernel này trong slide "Lecture04-CUDAParallelExecution(P2).pdf"; ở đây, để đơn giản, ta giả định *2*kích-*

thước-block = 2^k với k là một số nguyên dương nào đó). Hàm kernel sẽ tính giá trị tổng cục bộ trong từng block; sau đó, dùng hàm atomicAdd để cộng các tổng cục bộ này.

- Tính "gridSize" (từ "blockSize" và "n") khi gọi hàm kernel.

Với mỗi hàm kernel, chương trình sẽ in ra:

- Kích thước grid và kích thước block.
- Thời gian chạy của hàm kernel ("kernel time")
- "CORRECT" nếu kết quả tính được giống với kết quả đúng, "INCORRECT" nếu ngược lại.

Hướng dẫn về các câu lệnh (các câu lệnh dưới đây là chạy trên terminal của Linux, khi chạy ở code cell của Colab thì bạn thêm dấu ! ở đầu):

- Biên dịch file "HW2_P1.cu": nvcc HW2_P1.cu -o HW2_P1
- Chạy file "HW2_P1": ./HW2_P1
  Mặc định thì sẽ dùng block có kích thước 512; nếu bạn muốn dùng block có kích thước khác, chẳng hạn 256, thì bạn truyền thêm tham số dòng lệnh: ./HW2_P1 **256**

Báo cáo (2đ)

Bạn làm ở các mục "Câu 1A" và "Câu 1B" trong file "HW2.ipynb" mà mình đính kèm.

Câu 1A (1đ)

- Biên dịch và chạy file "HW2.cu" (để kích thước block mặc định).
- Từ kết quả chạy, bạn so sánh "kernel time" của 3 hàm kernel với nhau và thử giải thích xem tại sao lại như vậy (chỗ nào bạn không biết tại sao thì cứ nói là không biết tại sao).
- Dùng nvprof để xem chi tiết thực thi của chương trình. Có nhận xét gì về thời gian thực thi các công việc trong phần "GPU activities"

Câu 1B (1đ)

Chạy file đã biên dịch ở câu 1 với các kích thước block khác nhau: 1024, 512, 256, 128.

Ở câu này ta chỉ tập trung vào kết quả của hàm kernel 1. Với mỗi kích thước block, bạn điền các kết quả của hàm kernel 1 theo mẫu bảng biểu bên dưới. Với số block / SM và occupancy, ngoài việc điền kết quả vào bảng thì bạn cũng cần giải thích thêm là tại sao lại tính ra được như vậy. Khi tính số block / SM và occupancy, tạm thời ta chỉ xét 2 ràng buộc của SM là số block tối đa và số thread tối đa; bạn có thể tra cứu 2 ràng buộc này ở document của CUDA, mục "Programming Guide" ("Guide" không có s), mục "H. Compute Capabilities", bảng "Table 14. Technical Specifications per Compute Capability", 2 ràng buộc đó là "Maximum number of resident blocks per multiprocessor" và "Maximum number of resident threads per multiprocessor".

Block size	Grid size	Num blocks / SM	Occupancy (%)	Kernel time (ms)
1024				
512				
256				

Từ bảng biểu đã điền, bạn thử suy nghĩ và giải thích xem tại sao khi thay đổi block size thì "kernel time" lại thay đổi như vậy?

Câu 2 (6đ)

Áp dụng các kiến thức đã học về CUDA để cài đặt thuật toán nhân hai ma trận các số thực:

$$C\ (m \times k) = A(m\ \times n)\ .B(n\ \times k)$$

Code (4đ)

Mình đã viết sẵn cho bạn khung chương trình trong file "HW2_P2.cu" đính kèm; bạn chỉ viết code ở những chỗ có từ "// TODO" :

- Sinh viên cần cài đặt hai hàm kernel. Kernel1: Basic Matrix Multiplication, cài đặt đơn giản sử dụng global memory.
- Kernel2: TiledMatrixMultiplication. Cài đặt tối ưu hơn, sử dụng shared memory

Với mỗi hàm kernel, chương trình sẽ in ra:

- Kích thước grid và kích thước block.
- Thời gian chạy
- Sự sai khác giữa kết quả host và device. 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.

Hướng dẫn về các câu lệnh (các câu lệnh dưới đây là chạy trên terminal của Linux, khi chạy ở code cell của Colab thì bạn thêm dấu ! ở đầu):

- Biên dịch file "HW2_P2.cu": nvcc HW2_P2.cu -o HW2_P2
- Chạy file "HW2_P2": ./HW2_P2

Trong bài này, chúng ta sẽ tạm cố định kích thước block là 32 x 32

Báo cáo (2đ)

Bạn làm ở các mục "Câu 2" trong file "HW2.ipynb" mà mình đính kèm.

Câu 2A

- Biên dịch và chạy file "HW2.cu" (để kích thước block mặc định).
- Dùng nvprof để xem chi tiết thực thi của chương trình. Có nhận xét gì về thời gian thực thi các công việc trong phần "GPU activities"

Câu 2B

Sinh viên trả lời các câu hỏi trong file "HW2.ipynb", mục 2B

Nộp bài

Bạn tổ chức thư mục bài nộp như sau:

Thư mục <MSSV> (vd, nếu bạn có MSSV là 1234567 thì bạn đặt tên thư mục là 1234567)

- File code “HW2_P1.cu”
- File code “HW2_P2.cu”
- File báo cáo “HW2.ipynb”

Sau đó, bạn nén thư mục <MSSV> này lại và nộp ở link trên moodle.