

# Parallelize and optimize an application



KHOA CÔNG NGHỆ THÔNG TIN  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

**fit@hcmus**

# Parallelizing/optimizing a program

What part(s) should be parallelized/optimized?

- Measure times of parts to decide
- When optimizing (after parallelizing), can measure times of GPU activities quickly:  
`nvprof --print-gpu-trace ./a.out`

**1. Analyze**

**2. Design**

**3. Implement**

**4. Evaluate**

How to parallelize/optimize?

- Each loop will create a new version based on previous versions
- We should go step by step, from sequential to parallel, from parallel to optimized parallel

Does the idea work?  
If not, do you know why?

# Parallelizing/optimizing a program

What part(s) should be parallelized/optimized?

- Measure times of parts to decide
- When optimizing (after parallelizing), can measure times of GPU activities quickly:  
`nvprof --print-gpu-trace ./a.out`

**1. Analyze**

**2. Design**

**3. Implement**

**4. Evaluate**

How to go through this process as well as possible?

Some advices:

- **Keep the mind still**
- Keep the code clean
- Code fast or slow?
- Use a good editor and learn how to use it efficiently

How to parallelize/optimize?

Does the idea work?  
If not, do you know why?

# General optimization guidelines

- Expose enough independent tasks to utilize GPU hardware resources
  - ▣ Expose enough blocks to utilize SMs
  - ▣ In each SM, expose enough independent instructions (coming from the same warp, or from different warps) to utilize execution pipelines, hide latency
- Access DRAM efficiently
  - ▣ Don't let threads in the same warp access scattered addresses in DRAM
  - ▣ Use SMEM to reduce DRAM accesses, as well as to access DRAM efficiently
- Reduce warp divergence

# Final project - Contents of Colab notebook

## 1. Application description Viết mô tả về ứng dụng

❑ What is your chosen application?

- Input? Output?
- Use cases?

❑ Does it need to speed up?

Bài toán có đáng làm hay không

=> Nhấn mạnh Song song chạy nhanh hơn

Phải mô tả được input và output là gì? Đưa ra ví dụ:

=> Input là một bức ảnh và output sẽ là nhãn tương ứng

Use case: Ứng dụng này dùng trong ứng dụng gì?

Giải thích tại sao phải speed up thuật toán?

=> Trình bày tất cả vào file notebook!

# Final project - Contents of Colab notebook

## 2. Sequential implementation

- ❑ Design: Describe steps to go from input to output (don't show code)
- ❑ Evaluate:
  - Describe your experiment setup
  - Run the code to see results
  - Does it run correctly?

Mô tả thuật toán tuần tự chạy ra sao, nên dùng sơ đồ, chạy step-by-step, có thể dùng mã giả nhưng không show code

Đánh giá: Mô tả về thí nghiệm, chạy trên tập dữ liệu nào, thiết bị gì, có bao nhiêu dữ liệu huấn luyện, bao nhiêu dữ liệu test. Lấy mẫu có ngẫu nhiên không?

Xem thuật toán có chạy đúng hay không?

# Final project - Contents of Colab notebook

## 3. Parallel implementation

- ❑ Analyze: Which steps do you parallelize? Why these steps?
- ❑ Design: How do you parallelize? (don't show code)
- ❑ Evaluate:
  - Describe your experiment setup
  - Run the code to see results
  - Does it **run correctly** & **faster**? If not, do you know why?

Phân tích: Lý giải tại sao ta lại chọn song song ở chỗ nào? Tập trung vào song song hóa ở một số chỗ quan trọng, chỗ nào không quan trọng thì chạy tuần tự

Thiết kế:

Đánh giá: Mô tả setup, chạy code để xem kết quả, code chạy đúng không (so sánh với bản tuần tự), xem thời gian chạy. Các version có<sup>7</sup> thể có vấn đề nhưng ta sẽ giải quyết ở các version sau

# Final project - Contents of Colab notebook

## 4. Parallel implementation + optimization

You should have  $\geq$  2 optimized versions

At each version:

- ❑ Analyze: Which parts (often: which kernels) do you optimize? Why these parts?
- ❑ Design: How do you optimize? (don't show code)
- ❑ Evaluate:
  - Describe your experiment setup
  - Run the code to see results
  - Does it run correctly & faster? If not, do you know why?

Cải tiến các version khác đó. Cài đặt phiên bản tối ưu hơn của thuật toán song song.

Phân tích: chọn ra kernel cần giải quyết và thiết kế



# Final project - Contents of Colab notebook

## 5. Reflection

- ❑ Each member: What difficulties have you encountered?
- ❑ Each member: What have you learned?
- ❑ Your team: If you had more time, what would you do?

Ghi nhận lại các khó khăn đã gặp, mỗi người đã học được những gì, nếu có thời gian thì nhóm sẽ làm thêm điều gì?

# Final project - Contents of Colab notebook

## 6. References

To finish this project, what materials have you consulted?

Phải viết các references vào file report Colab

# Final project - Code files

Each version (sequential version, parallel version, 1<sup>st</sup> optimized parallel version, 2<sup>nd</sup> optimized parallel version, ...) should be in a separate file

Có một file note ra phiên bản nào sẽ có các file nào

# Final project - Teamwork

Your team should have a plan file

All members in your team should understand the team's project thoroughly (of course, it includes code)

Viết kế hoạch thực hiện

# Final project - Submission & presentation

- ☐ x = presentation day  
x will be one day from xxx to xxx (I will decide and let you know later)
- ☐ **Before 23:55 day x-1:** upload your team's project to a link in Moodle, include:
  - ☐ Team plan file and work distribution
  - ☐ Colab notebook file
  - ☐ All source code file and an instruction file on how to set up and run your project
  - ☐ A presentation video about 15-20min. Upload on YouTube with Unlisted option
- ☐ **Day x:** present offline in classroom (use Colab notebook file to present, no need to prepare slides)
  - Each team will have ~15 minutes to present (each member will present ~1/2 contents, and I will decide who will present which) and ~10 minute to Q & A
  - ~ 30 min
  - 1 cột điểm  
Thầy TH

Trước buổi vấn đáp, cần up bài nộp lên Moodle

*Thank you*

