Parallelize and optimize an application

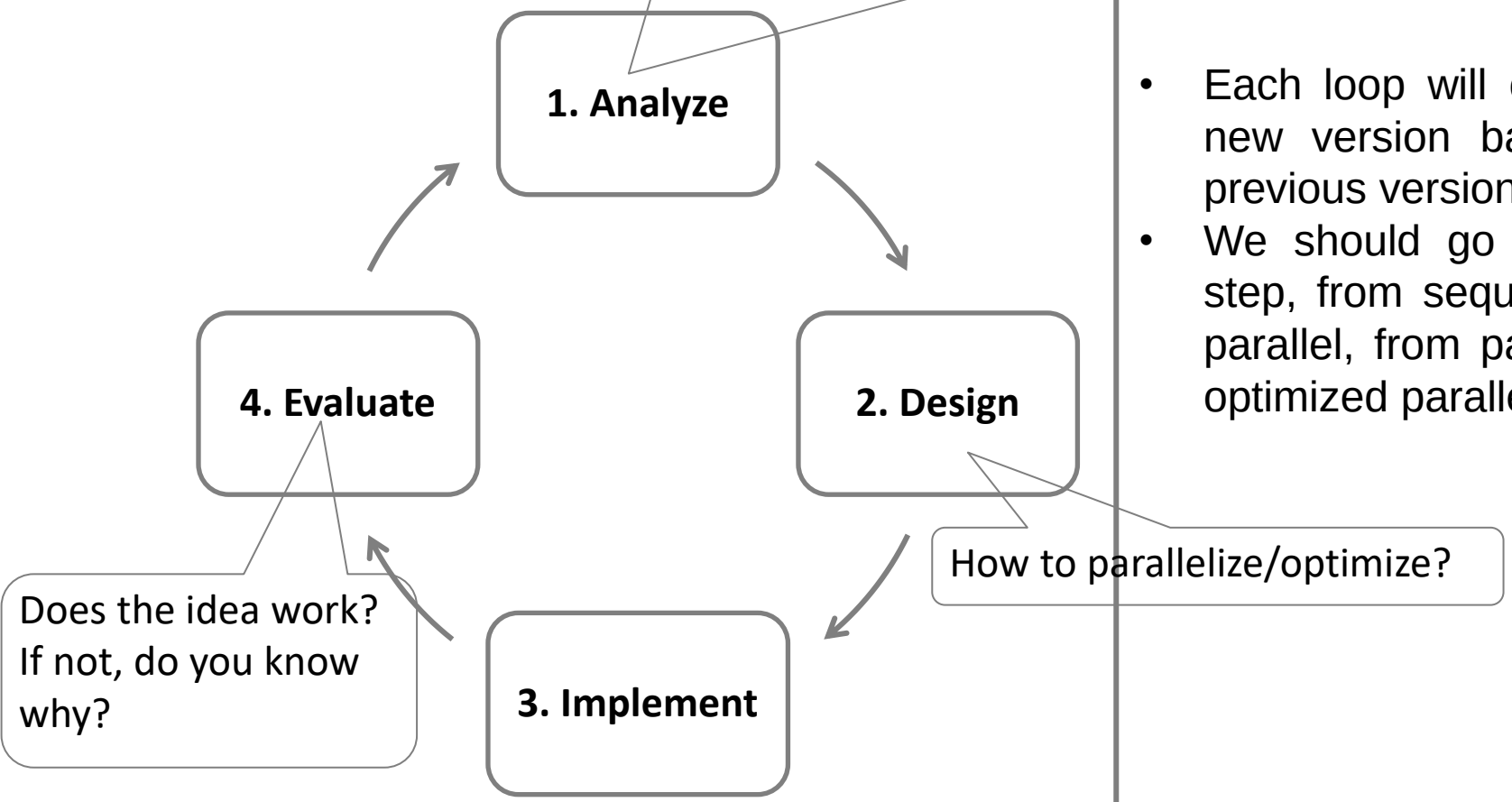
KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

fit@hcmus

Parallelizing/optimizing a program

What part(s) should be parallelized/optimized?

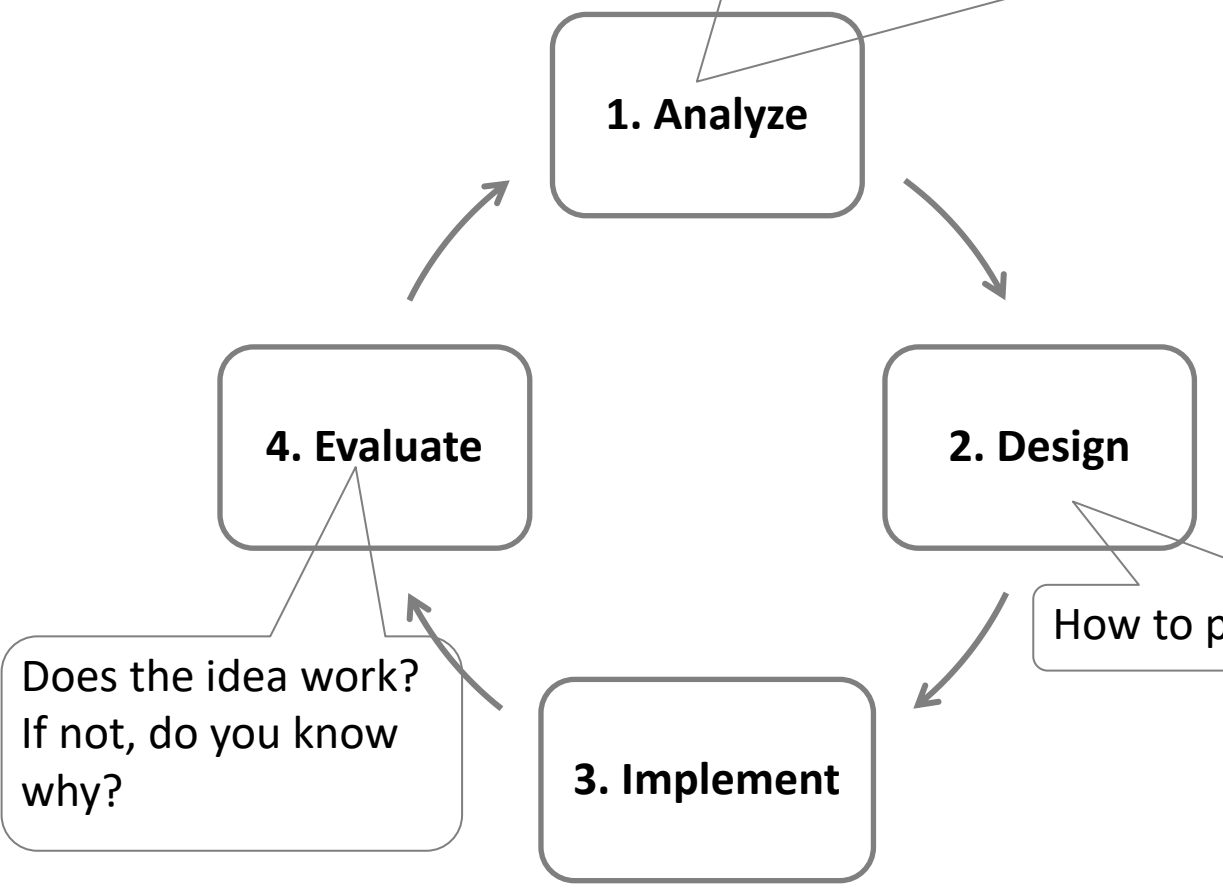
- Measure times of parts to decide
- When optimizing (after parallelizing), can measure times of GPU activities quickly:
  nvprof --print-gpu-trace ./a.out



- 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

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



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

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

Parallelize and optimize an application

(you can choose your own application or my suggested application)

Final project - Contents of Colab notebook

1. Application description

- What is your chosen application?
  - Input? Output?
  - Use cases?
- Does it need to speed up?

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?

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?

Final project - Contents of Colab notebook

4. Parallel implementation + optimization

You should have ≥ 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?

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?

Final project - Contents of Colab notebook

6. References

To finish this project, what materials have you consulted?

Final project - Code files

Each version (sequential version, parallel version, 1st optimized parallel version, 2nd optimized parallel version, …) should be in a separate file

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)

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
  - Colab notebook file
  - Code files
- **Day x:** present via zoom (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

My suggested app to parallelize and optimize:

image resize using seam carving

1. Application description

- **Input:** an image (this project: RGB image)
- **Output:** the resized image without distorting important objects (this project: the resize image with smaller width, same height)

- One possible use case: we may want display an image on different devices (computer, cell phone, …)
- Does this app need to speed up?

  Yes, it's slow!

Original image

Scaling

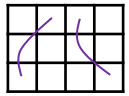
Cropping

Seam carving

Image source: https://en.wikipedia.org/wiki/Seam_carving

2. Sequential implementation (idea)

A seam: a set of pixels, one pixel per row,
pixels of row r and r+1 are connected



Find the least important seam → Remove this seam → Find the least importa …



Repeat this process until achieving the desired width

Demo …

Find the least important seam

Find the importance of each pixel
(One approach: use edge detection)

Find the least important seam
from pixel importances

Find the importance of each pixel

One approach: use edge detection

- Convert RGB image to grayscale image
- Detect edges in the x-direction (1): do convolution of this grayscale image with x-Sobel filter

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

- Detect edges in the y-direction (2): do convolution of this grayscale image with y-Sobel filter

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- Importance of a pixel = |the corresponding result of (1)|
  + |the corresponding result of (2)|

Find the least important seam from pixel importances

Compute pixel-importance of all possible seams?

$O(\text{width} \times 3^{\text{height}})$ ☹

1	4	3	5	2
3	2	5	2	3
5	3	4	2	1

Pixel importance

Find the least important seam from pixel importances

A O(width × height) way
of finding the seam with least pixel-importance

1	4	3	5	2
3	2	5	2	3
5	3	4	2	1

Pixel importance

Least pixel-importance to bottom

Find the least important seam from pixel importances

A O(width × height) way
of finding the seam with least pixel-importance

1	4	3	5	2
3	2	5	2	3
5	3	4	2	1

Pixel importance

5	3	4	2	1

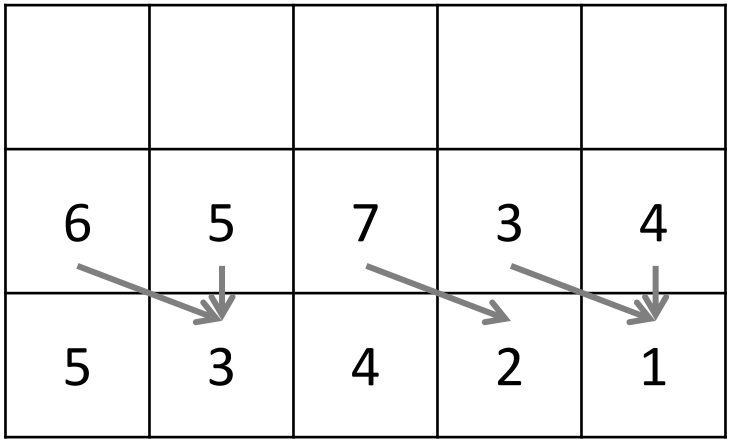
Least pixel-importance to bottom

Find the least important seam from pixel importances

A O(width × height) way
of finding the seam with least pixel-importance

1	4	3	5	2
3	2	5	2	3
5	3	4	2	1

Pixel importance



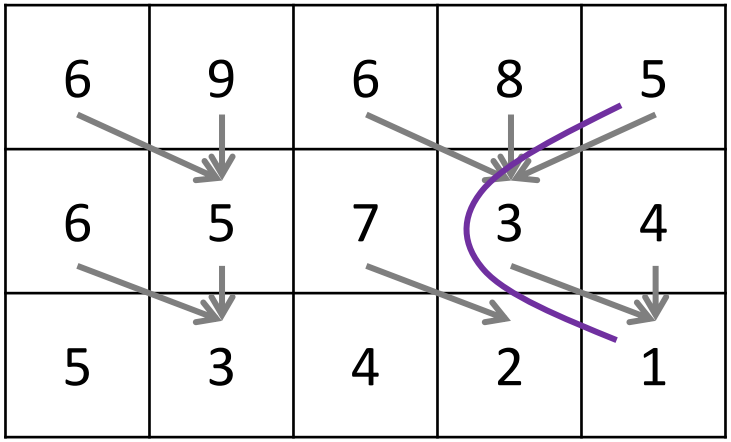
Least pixel-importance to bottom

Find the least important seam from pixel importances

A O(width × height) way
of finding the seam with least pixel-importance

1	4	3	5	2
3	2	5	2	3
5	3	4	2	1

Pixel importance



Least pixel-importance to bottom

Seam carving references

(Terminology: “energy” = importance)

- Shai Avidan & Ariel Shamir, Seam Carving for Content-Aware Image Resizing (original paper)
- 18.S191 MIT Fall 2020, Seam Carving: part1, part2
- Wikipedia, Seam Carving

Thank you