Parallelize and optimize an application

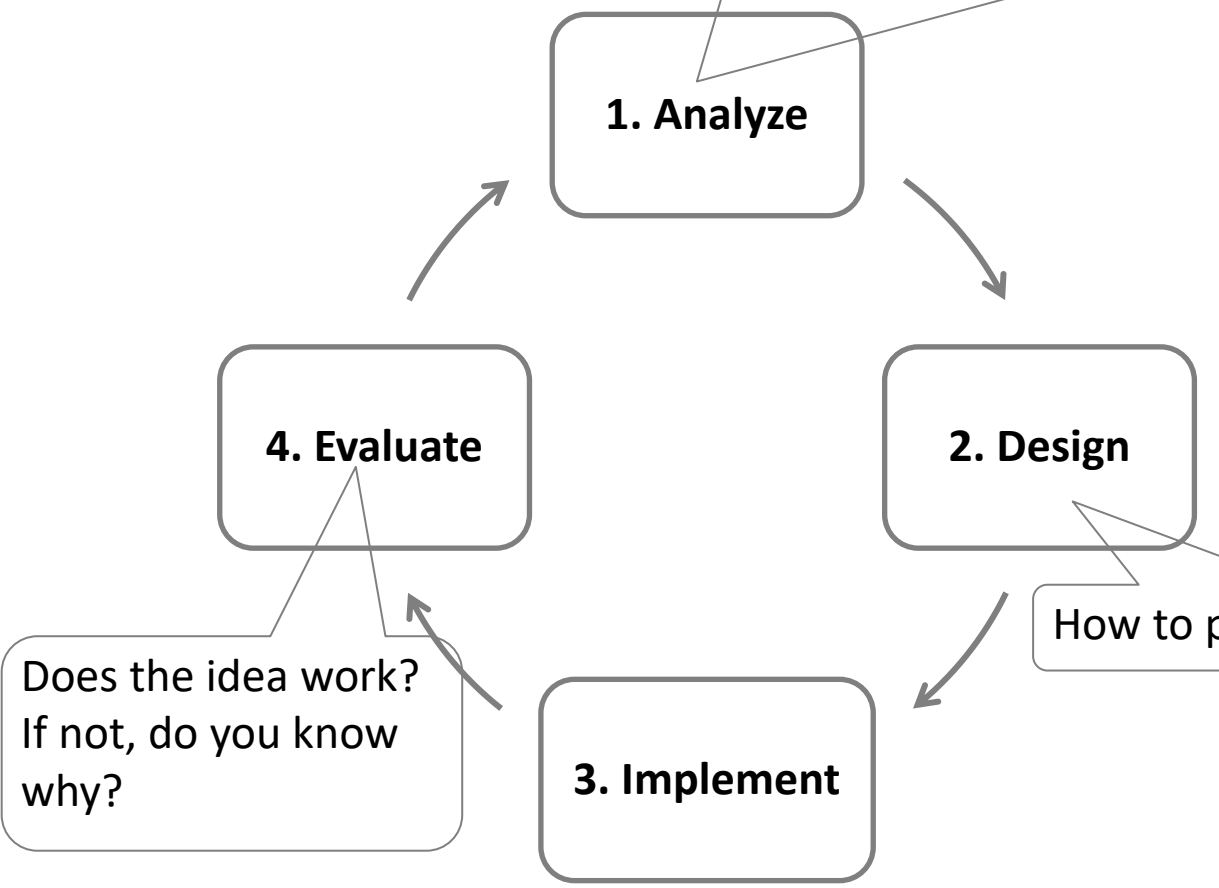
KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

fit@hcmus

Parallelizing/optimizing a program

What part(s) should be parallelized/optimized?

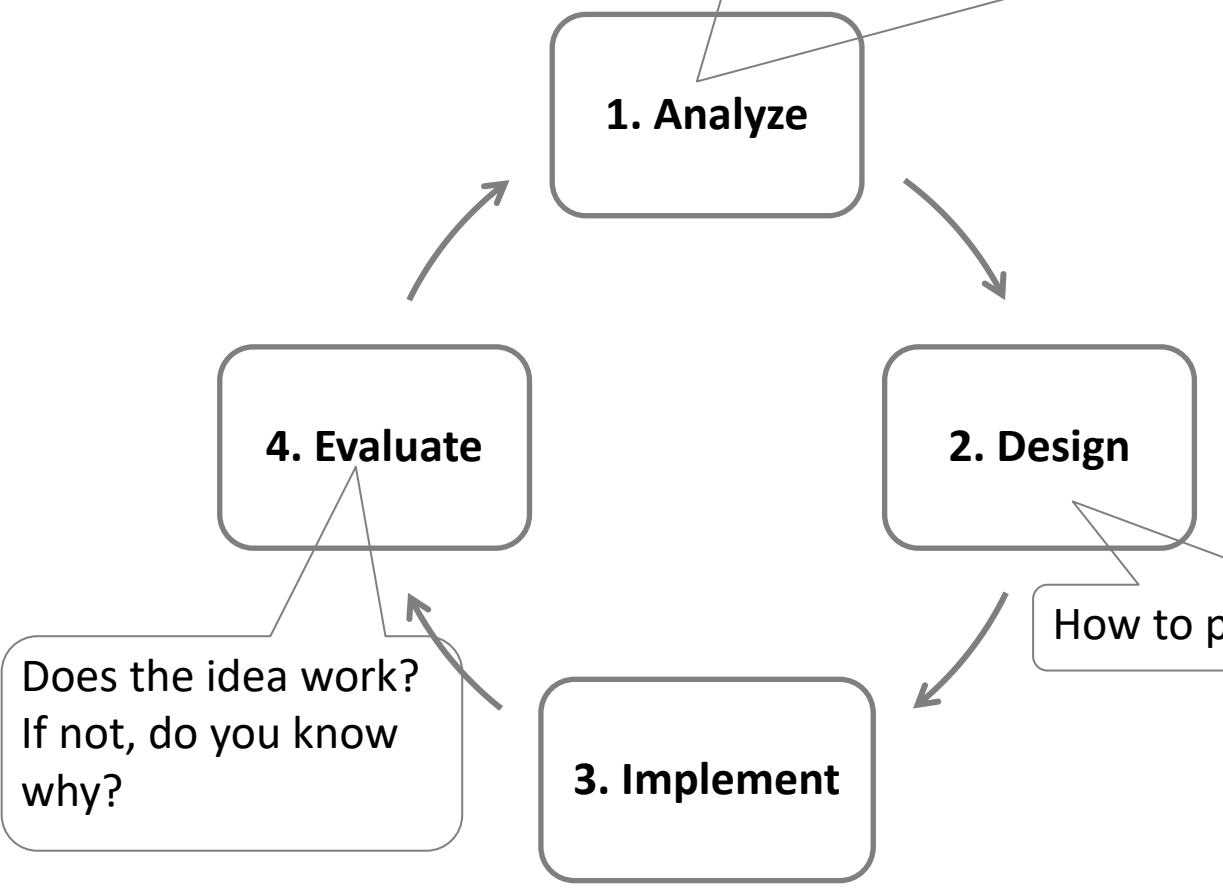
- Measure times of parts to decide
- When optimizing (after parallelizing), can measure times of GPU activities quickly:
  nvprof --print-gpu-trace ./a.out



- 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 - 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 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

Thank you