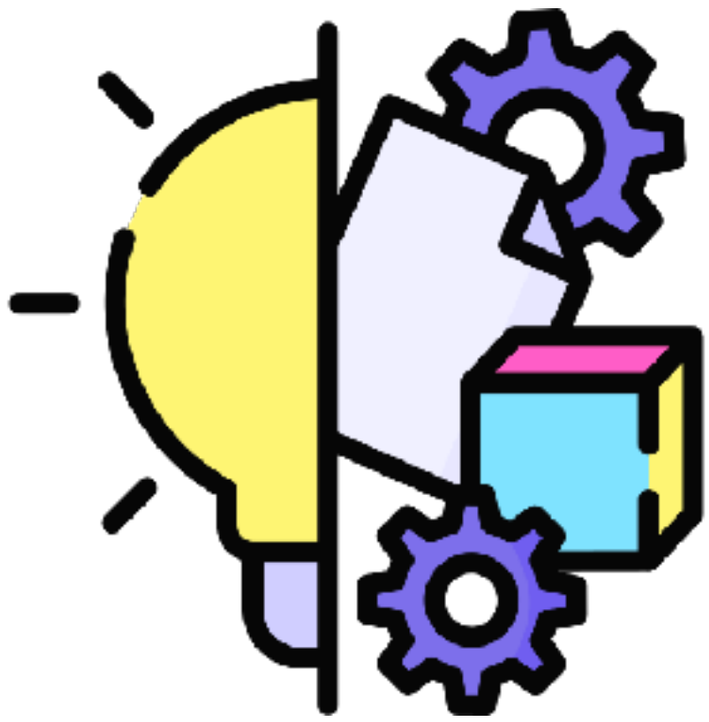

Functional and Parallel Programming:

Final Project (10%)

Overview & Objectives

Objectives:

- To delve deeper into topics connected to the theme of the class (parallel, concurrent, functional)
- To brag about speed, coolness & awesomeness of what you'll be able to accomplish
- To learn a lot and have fun either by yourself or with friends (Teams of ≤ 2)

Duration: Now until the final exam week

Deliverables:

- a brief proposal (due next Tue)
- a GitHub repo with your project + a short writeup summarizing the work
- a science-fair-style poster (presented in the FE slot)

Sample Ideas

- Speed up a process/algorithm you care about by making it parallel and/or taking advantage of concurrent data structures. Your writeup will summarize what has been done and the benefits gained. E.g., make BFS really fast

- Implement something that compares approaches to parallelism/ concurrency in Rust/Scala (e.g., Actor vs. Future vs. pure Threading)

- Implement an image processing program (e.g., fun filters)

- Implement a simple parallel database supporting basic select queries.

- Beat numpy or scikitlearn and create a Python wrapper for it.

Project Proposal (3%)

- No more than one A4 page
- Clearly define what you will work on. For teams with more than one member, what each person will work on.
- Indicate how you'll measure success.
- Remember: use this project as an opportunity to learn something new and have fun.
- Litmus test: the amount of work should be ~ if you spend 1-1.5 hours everyday on this for the rest of the term.

Project Submission (7%)

- A short write-up detailing what you've done in PDF or Markdown.
- Commit early and commit often. We'll look at your commits.
- Everything but (gigantic) datasets should be preserved on there. You can decorate your resume with it.
- Due: one week after last day of FunPar lecture