Milestone Report

Group Members:

Xingyuan Wang (xingyuaw@andrew.cmu.edu) Hangrui Cao (hangruic@andrew.cmu.edu)

Following Schedule for the Project

The current and updated schedule track for this project is as follows.

Week	Dates	Tasks Completed/Planned
0	March 25 – March 31	Reviewed several existing AMR codebase and literature. (both)
1	April 1 – April 7	Further reviewed a structured AMR repository that simulates heat propogation but does not modify mesh during execution. (Xingyuan Wang) Examined wave simulation problem and finalized data structures to use. (Hangrui Cao)
2	April 8 – April 14	Implemented grid initialization and a (simplified) workload rebalancing algorithm. (Xingyuan Wang) Implemented and tested flux calculation procedure; implemented mesh refinement criteria. (Hangrui Cao)
3	April 15 – April 17	Finish mid-project report. (both) Fix bugs when program runs near grid boundaries, and complete the initial solution. (Xingyuan Wang) Implement visualization script for displaying wave grid. (Hangrui Cao)
3	April 18 – April 21	Run initial performance tests on GHC machines with varying processor numbers, identiy any performance issues. (Xingyuan Wang) Test another workload rebalancing algorithm with finer granularity. (Hangrui Cao)
4	April 22 – April 24	Support for multi-level refinement (if possible, currently fixed at 2). (Xingyuan Wang) Examine locality and MPI communication schema issues. (Hangrui Cao)
4	April 25 – April 28	Refine current memory access pattern with improved data layout. (Hangrui Cao) Get final performance results from GHC and PSC machines. (Xingyuan Wang)

Week	Dates	Tasks Completed/Planned
5	April 29 – May 5	Final comparison tests, make project poster and write the final report. (both)

Current Progress

At this midpoint milestone stage, the progress we have made based on our schedule includes:

- **Diving Deep into the AMR (Adaptive Mesh Refinement) Problem**: We have thoroughly explored the complexities and challenges associated with Adaptive Mesh Refinement, a technique used to increase the resolution of simulations dynamically where it is most needed. This deep dive involved understanding the mathematical and computational foundations required to effectively implement AMR. For Adaptive mesh refinement, it has use cases in a lot of physics simulation, and we have picked the fluid simulation (shallow water simulation) as the use case for the AMR[1][2], which depends on some differential equations to calculate the shallow water flow with AMR. For the problem, the parallelization parts would have some **difficulty**, especially for the load-balancing part.

- **Serial AMR Version Implementation**: Following our research and exploration of AMR, we successfully implemented a serial version of AMR. This version serves as a foundational model that operates without parallel processing. The development of this serial implementation allowed us to address fundamental aspects of the refinement process, setting the stage for more complex, parallel implementations in the future.

What we are doing

To achieve our goals and deliverables, we are currently working on things based on our schedule, mainly including the parts in literature review, system formulation, basic understanding the AMR problem and implementation of the basic serial version of the AMR problem. We are currently working on implementing the AMR with MPI and we believe ourselves are on the right track of the project.

Plan To Show

As our project is adaptive mesh refinement, to illustrate the process of the adaptive mesh refinement, we want to show the project in following ways:

- **Graph**: We want to use graphs to represent some data in terms of cache line, locality, etc. Also, graphs will be used for comparison between parallel and sequential approaches. We also want to provide a simple dashboard (possibly supported by python library such as streamlit).

- **GIF or Short Video**: We want to show under different setting, the dynamic movement of the meshes before and after the refinement.

Preliminary Results

Currently, we have finished the serial version implementation of the AMR code for this physic phenomenon of water flow (the problem is structured as placing the blocks in a square domain for simplicity). Currently, we are working on implementing the OpenMP version of the AMR.

Concern Problems

The problem we currently concern the most is the difficulty of the OpenMP implementation, as the load balancing part in this project is really hard to handle. We are working on implementing a very basic OpenMP version at first.

References:

[1] Ambrosi, D. (1995). Approximation of shallow water equations by roe’s riemann solver. International journal for numerical methods in fluids, 20(2):157–168 [2] Antepara, O., Lehmkuhl, O., Borrell, R., Chiva, J., and Oliva, A. (2015). Parallel adaptive mesh refinement for large-eddy simulations of turbulent flows. Computers & Fluids, 110:48–61.