APMA 2822B Final Project Proposal

Yash Agrawal

Problem Statement

The goal of this project is to write a highly parallelized 3D heat equation solver.

Equation

The 3D heat equation I will be solving is given by:

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right)$$

where $u(x, y, z, t)$ represents the temperature at point (x, y, z) at time t, and α is the thermal diffusivity constant. α will be set to 1 for simplicity.

The domain will be defined on $(x, y, z) \in [0, 1]^3$, or the unit cube.

I will test using the following initial condition:

$$u_0(x, y, z) = \sin(\pi x) \sin(\pi y) \sin(\pi z)$$

This gives the analytical solution:

$$u(x, y, z, t) = e^{-3\pi^2 t} \sin(\pi x) \sin(\pi y) \sin(\pi z)$$

I will use the Dirichlet boundary conditions, where the temperature on the boundary of the cube is held constant at zero:

$$u(x, y, z, t) = 0 \quad \text{for} \quad x = 0,\, x = 1,\, y = 0,\, y = 1,\, z = 0,\, z = 1$$

For updating the temperature at each time step, I will use explicit time stepping with the forward Euler method, defined as:

$$u^{n+1} = u^n + \Delta t \alpha \nabla^2 u^n$$

To discretize the domain, I will use a uniform grid with spacing $\Delta x = \Delta y = \Delta z = h$ for simplicity. Each dimension will be decomposed into N points (so $N - 1$ segments), where $h = \frac{1}{N-1}$.

For the time steps, I will choose Δt as follows to ensure stability:

$$\Delta t = 0.5 \cdot \frac{h^2}{6}$$

Implementation Plan

I plan to parallelize my implementation of the heat diffusion simulation using MPI and GPU acceleration with HIP.

MPI

I will use MPI to decompose the 3D domain. I will attempt to decompose all three dimensions to ensure true scalability with respect to N, so that no matter the size of N, increases the number of processes can give each process a reasonable number of points to compute.

I plan to use halo exchange to handle communication between neighboring subdomains.

GPU Acceleration

I will implement the core computation of the heat equation on the GPU using HIP. Each GPU thread will be responsible for updating the temperature at a single grid point. MPI will be used to ensure the number of points each process has is reasonable so that this strategy is effective.

Evaluation

I will evaluate my implementation on various grid sizes.

I will use my selected initial condition and analytical solution to evaluate the accuracy of my implementation, ensuring that the numerical solution converges to the analytical solution with some reasonable error tolerance.

I plan to generate a roofline model for the core computation to evaluate the performance of my implementation.

I will also attempt to benchmark specific GPU operations like memory allocations, data transfers, and kernel execution times.

Reach Goals

I have a couple reach goals in mind in case I manage to complete the core project early.

First, I want to attempt to visualize the 3D heat diffusion simulation using a library like VTK. This would involve needing to move the data from the GPU back to the CPU on every step (or few steps) which is a significant change. Ideally, I would be able to generate a roofline model for this version as well and observe the performance impact of the data transfers.

If I can complete this, I would also like to add a source term to the heat equation to generate more interesting simulations.

I potentially want to explore moving beyond one thread per grid point, and will consider implementing shared memory tiling if I have time.