Week 11 exercises

MATH 471/571 - Parallel Scientific Computing

Michal A. Kopera

Finite differences on GPU

We will revisit 1D finite difference code we have been working on at the beginning of the semester. Just to remind you, the finite difference method allows us to approximate the value of a derivative by computing:

$$\frac{\partial u}{\partial x} \approx \frac{u_{i-1} + u_{i+1}}{2\Delta x}.$$

Your task will be to convert the 1D finite difference code attached in the finite_difference.c file to CUDA code by writing required kernels. Please complete the following tasks:

Task 1 - Create finite difference kernel

The structure of the CPU code looks as follows:

1. Allocate memory
2. Create initial condition
3. Compute derivative
4. Report result / write to file

Incorporate the CUDA programming model to offload the derivative computation to the GPU:

1. Allocate device memory
2. Move data to device
3. Call finite difference kernel on the device
4. Copy the result to host
5. Deallocate device memory

You will have to:

a) save the finite_difference.c code using .cu extension

b) add steps 1,2,4,5 to your code

c) create finite difference kernel, which will replace step 3 - Compute derivative in your original code.

d) Make sure you get correct result by comparing either to the original CPU solution, or to the exact solution

Task 2

Moving data to the device can take significant time. In this example, we can afford to (partially) avoid this step by generating the initial condition directly on the GPU. You will still need to send the size of the array.

Create a kernel which will replace the init_u function (Step 2) and remove the “Move data to device” step from your CUDA code. Make sure you still get a correct result. Save this code as another .cu file, as you will need both for Task 3.

Task 3

Time the serial, MPI, and both CUDA codes from Tasks 1 and 2 for a few different values of problem size N. For the MPI code choose 28 processes (a full node). How do the times compare. Make sure that at least one of the problems is big enough to show significant differences between the implementations.