P. Hadjidoukas

Set 7 - MPI I

Issued: May 3, 2023

Question 1: 2D Diffusion and MPI

Heat flow in a medium can be described by the diffusion equation

$$\frac{\partial \rho(\boldsymbol{r},t)}{\partial t} = D\nabla^2 \rho(\boldsymbol{r},t), \tag{1}$$

where $\rho(\boldsymbol{r},t)$ is a measure for the amount of heat at position $\boldsymbol{r}$ and time t and the diffusion coefficient D is constant.

Let's define the domain Ω in two dimensions as $\{x,y\} \in [-1,1]^2$. Equation 1 then becomes

$$\frac{\partial \rho(x,y,t)}{\partial t} = D\left(\frac{\partial^2 \rho(x,y,t)}{\partial x^2} + \frac{\partial^2 \rho(x,y,t)}{\partial y^2}\right). \tag{2}$$

Equation 2 can be discretized with a central finite difference scheme in space and explicit Euler in time to yield:

$$\frac{\rho_{r,s}^{(n+1)} - \rho_{r,s}^{(n)}}{\delta t} = D\left(\frac{\rho_{r-1,s}^{(n)} - 2\rho_{r,s}^{(n)} + \rho_{r+1,s}^{(n)}}{\delta x^2} + \frac{\rho_{r,s-1}^{(n)} - 2\rho_{r,s}^{(n)} + \rho_{r,s-1}^{(n)}}{\delta y^2}\right) \tag{3}$$

where $\rho_{r,s}^{(n)} = \rho(-1 + r\delta x, -1 + s\delta y, n\delta t)$ and $\delta x = \frac{2}{N-1}$, $\delta y = \frac{2}{M-1}$ for a domain discretized with $N \times M$ gridpoints.

We use open boundary conditions

$$\rho(x,y,t) = 0 \quad \forall\, t \geq 0 \,\text{and}\, (x,y) \notin \Omega \tag{4}$$

and an initial density distribution

$$\rho(x,y,0) = \begin{cases} 1 & |x,y| < 1/2 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

a) Implement the OpenMP parallelization of the 2D diffusion equation. Parallelize the routines that initialize and advance the system.

b) Implement the MPI parallelization of the 2D diffusion equation by filling in all parts of the code marked by `TODO:MPI`. Decompose the domain using tiling decomposition scheme (described in the lecture notes). (i.e. distribute the rows evenly to the MPI processes).
   - *Note 1:* Study and become familiar with the provided OpenMP version of the code.
   - *Note 2:* Do not use non-blocking communication (which has not been discussed yet).

c) Compute an approximation to the integral of ρ over the entire domain in `compute_diagnostics`. Compare your result after 1000 iterations using the result of the provided OpenMP code that solves equation (1). To run the code use the parameters in Table 1.

d) Suggest other ways to divide the real-space domain between processes with the aim of minimizing communication overhead. Prove your argument by computing the message communication size for the tiling domain decomposition and for your suggestion.

e) (Optional) Make a strong and weak scaling plot.

Table 1: Example parameters.

		$\Omega : [-L, L]$	$N \times N$	timesteps	
	D	L	N	T	Δt
Set 1	1	1	128	1000	0.00001
Set 2	1	1	256	1000	0.000001
Set 3	1	1	1024	1000	0.00000001