Homework 7: Poisson

Han Tran

This code solves (approximately) the Poisson equation in 3D domain:

$$-\Delta \mathbf{u}(\mathbf{x}) = \sin(2\pi x) sin(2\pi y) sin(2\pi z) \quad \forall \mathbf{x} \in \Omega, u(\mathbf{x}) = 0 \text{ on } \partial\Omega.$$

The approximation of the Laplace operator employs the following equation

$$\frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{h^2} + \frac{u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{h^2} + \frac{u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k-1}}{h^2} = f_{i,j,k}$$

where $f_{i,j,k}$is the loading function. For this assignment, $f_{i,j,k} = \sin(2\pi x_i)\sin(2\pi y_j)\sin(2\pi z_k)$.

Using Jacobi iterative method, we have can compute the value of u in terms of the previous step in the iteration, i.e.

$$u_{i,j,k}^{(l+1)} = \frac{1}{6}\left(u_{i-1,j,k}^{(l)} + u_{i+1,j,k}^{(l)} + u_{i,j-1,k}^{(l)} + u_{i,j+1,k}^{(l)} + u_{i,j,k-1}^{(l)} + u_{i,j,k+1}^{(l)} - h^2 f_{i,j,k}\right)$$

where super indices (*l*+1) and (*l*) denote the current step and previous step respectively, *h* is the grid size, $h = \frac{L}{N+2}$, *L* is the length per dimension (x, y and z) of the domain, *N* is the number of interior points per. For this assignment, *L* = 1000, *N* will be taken with various values to analyze the strong and weak scalability as explained below.

For convergence study, a relative error is computed as

$$e = \left\|u^{(l+1)} - u^{(l)}\right\|_2 = \left(\sum_{i=1}^{N+2}\sum_{j=1}^{N+2}\sum_{k=1}^{N+2}\left(u_{i,j,k}^{(l+1)} - u_{i,j,k}^{(l)}\right)\right)^{1/2}$$

When the relative error is less than a prescribed tolerance ($e < \varepsilon$), the iteration will exit. For this assignment, a value of $\varepsilon = 0.01$ is chosen.

For the analysis of strong scalability, a fixed value of N = 200 is taken. The problem runs on kingspeak.chpc.utah.edu using increasing number of processes as shown in the following table.

# nodes	total # processes	# processes per dimension	# interior points per dimension	# iterations	time taken
6	1	1	200	1056	959.613932
6	8	2	200	1056	139.79427
6	27	3	200	1056	108.967838
6	64	4	200	1056	47.939908
6	125	5	200	1056	26.329587

It can be seen that the program is strong scalable, i.e. the computing time decreases when the number of processes increases.

For the analysis of weak scalability, the ratio of N/p is kept to be constant (for this assignment $N/p = 100$). The computing time is shown on the following table.

# nodes	total # processes	# processes per dimension	# interior points per dimension	# iterations	time taken
6	1	1	50	56	1.138589
6	8	2	100	66	3.234682
6	27	3	150	41	2.015043
6	64	4	200	1056	47.965956
6	125	5	250	2556	115.460691

It can be seen that for the first 3 cases ($p = 1$, 8 and 27) the time is constant. However when the number of processes becomes much bigger ($p = 64$ and 125), the time increases. This seems that the code is not quite weak scalable. However, the chosen grain size is quite small (which is 50). This makes the computation (at each local process) versus the communication between processes are not well balanced, particularly when the number of processes increases to a very large number. I believe that when the grain size is bigger, the weak scalability could become more clearly. However, I cannot make the grain size bigger because of the limit of the memory of kingspeak and it can take much longer to run the experiments.