Documentation for the code and sample results

Code description

The code solves Boussinesq equations in 360^3 grid points using MPI parallelisation. The time step size $dt = 0.01$ and the code will be executed for total time unit of $T = 0.2$ such that the total number of time steps will be $T/dt = 20$. A detailed step by step procedure for running the code is given below. The total wall clock time taken by the code will be saved in a file named *Total_ Time_ taken.txt*.

Steps to run the code and get the scaling

- To use mpi4py package, install *intelpython* and *intel MPI*.
- Create 20 directories with names t_0, t_1 and so on up to t_20 such that data in each time will be saved to the respective directories. Note that the code saves data in each time step and hence number of directories should be same as the number of time steps.
- In order to change the total number of time steps, change the value of T in the code such that the total time step will be decided by T/dt. Note that the number of directories should be same as the number of time steps.
- A sample shell script is also provided in the folder (*create_ directories.sh*) which can be used to create 20 directories by executing it from the terminal as *./create_ directories.sh*.
- Each test result should be obtained by averaging results from 5 independent trials. A sample scaling table is provided below in Table 1 and a scaling curve obtained from the same data is given in Figure 1.

Number of cores	**Average time for 20 time steps**	**Average time for 1 time step**
90	160 s	8.0 s
120	128 s	6.4 s
180	86 s	4.3 s
360	46 s	2.3 s

Table 1

- **The lowest time taken for one time step, obtained from the right most column of Table 1, is defined as the *best time* and will be used to calculate expenses. For the sample scaling given in Table 1 the *best time* is 2.3 seconds and this used 360 cores.**

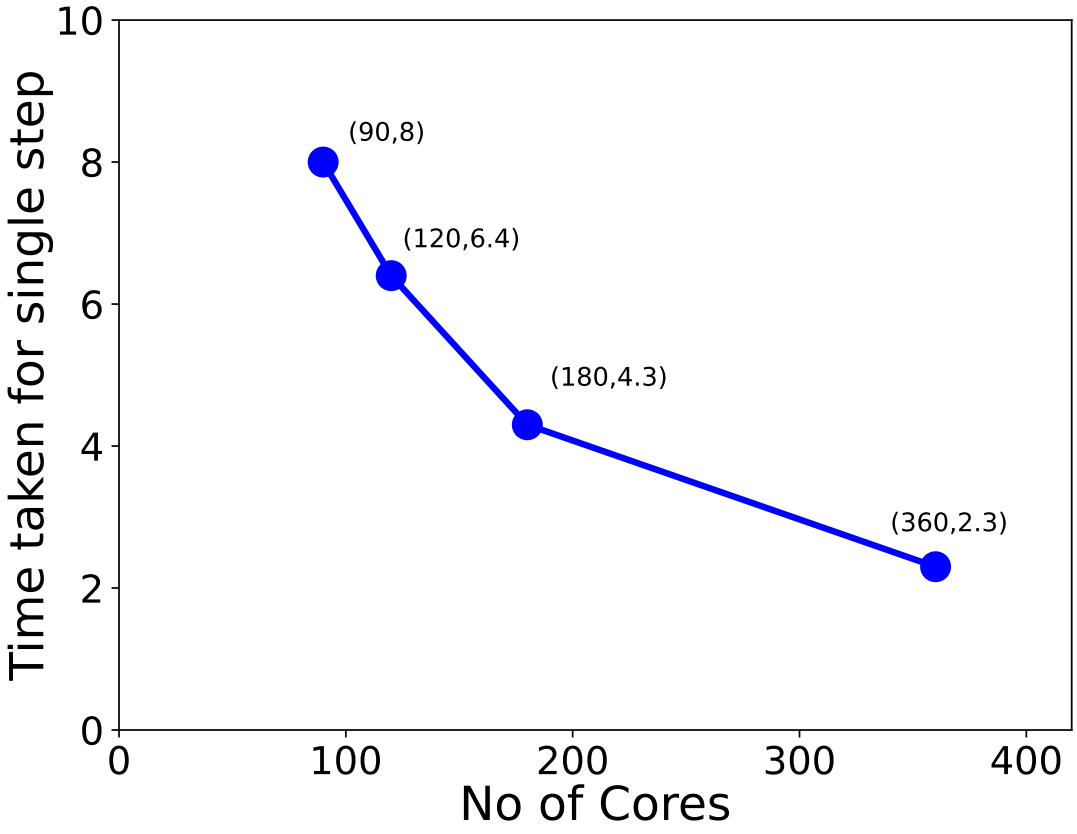
Time taken for single step
No of Cores
(90,8)
(120,6.4)
(180,4.3)
(360,2.3)
0
2
4
6
8
10
0
100
200
300
400

Figure 1