A simple script for testing programs

Testing programs' execution time is a tedious work. Here I'd like to share a python script to simplify this procedure.

Basically, it can run MPI or Pthread programs for specified times, and output the execution time to a `.txt` file. So when you start to analyze the execution time, you can directly copy the data in each log file. This script is tested on our cluster only (python 2.7), be careful whenever you use it.

How to use

1. Modify your program

In both `YOUR_MPI_PROGRAM.c` and `YOUR_PTHREAD_PROGRAM.c`, add the following codes to output the execution time:

```
        /* save execution time log */
        FILE *log_file_p;
        char file_name[100];
        sprintf(file_name, "MPI_%d_processes_execution_time.txt", num_process);
// you can change the file name format here
        log_file_p = fopen(file_name, "a");  // add a new log
        fprintf(log_file_p, "%f\n", total_time);
        fclose(log_file_p);
```

Notice: you should define the execution time yourself.

In `YOUR_PTHREAD_PROGRAM.c`, add the following codes so that the program can create pthreads with specific amount:

```
int NUM_PTHREADS;  // at very beginning
...
sscanf(argv[1], "%d", &NUM_PTHREADS);  // pass the amount of pthreads
...
pthread_t *threads = malloc(sizeof(pthread_t) * NUM_PTHREADS);
```

2. Execute

Before you run MPI program, you should load the module:

```
$ module load openmpi-3.1.2-gcc-8.2.0-qgxgzyn
```

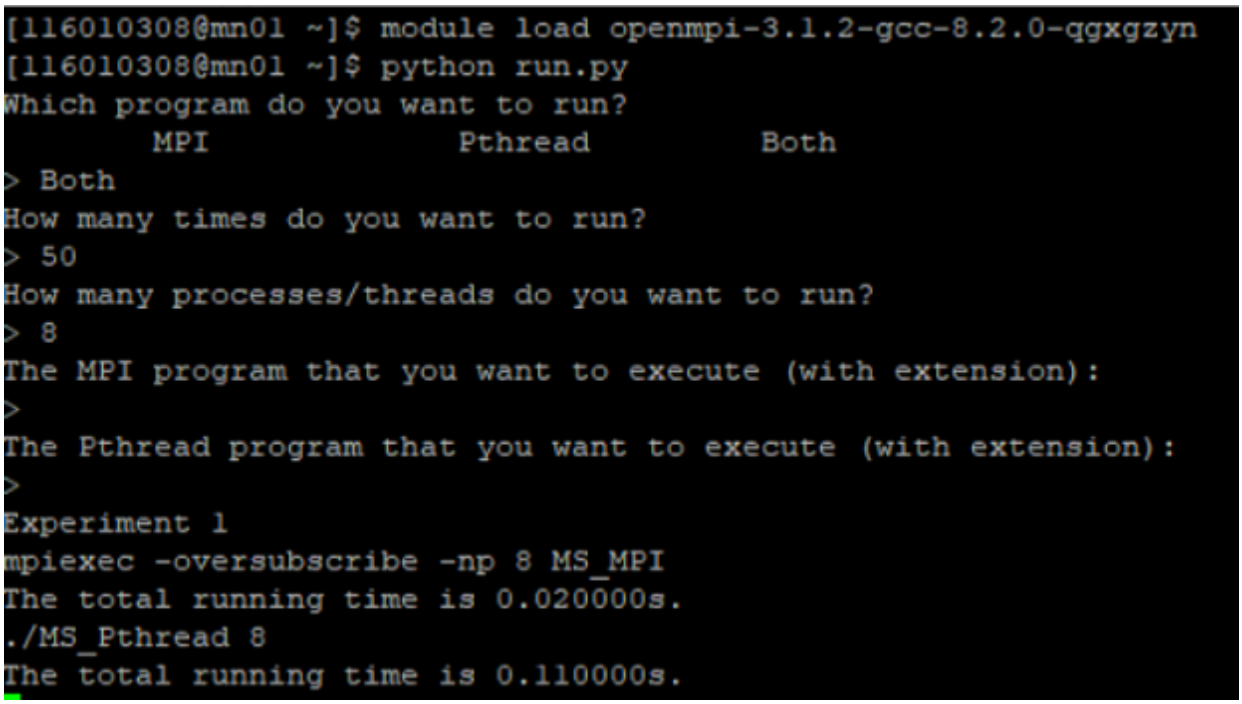
Then, copy this script to the directory of your programs. For example, my directory looks like:

```
.
├── MPI_mandelbrot_set.c
├── Pthread_mandelbrot_set.c
├── run.py
└── seq_mandelbrot.c
```

In cluster's terminal, run the command:

```
$ python run.py
```

Sample



You can check the number of available cores by `pestat` .