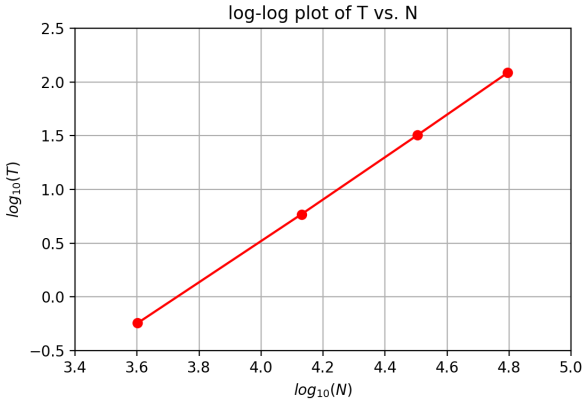
Assignment01

From Siqi Liang (liangsiq@usc.edu)

Part I

I-1. Measuring Computational Complexity



Code for plotting:

```
from matplotlib import pyplot as plt
import numpy as np
# data processing
data_path = './MDtime.txt'
data = np.loadtxt(data_path)
N_data = data[:,0]
log10N = np.log10(N_data)
T_data = data[:,1]
log10T = np.log10(T_data)

# plot the figure
plt.plot(log10N, log10T,'ro-')
plt.xlabel('$log_{10}(N)$')
plt.ylabel('$log_{10}(T)$')
plt.title('log-log plot of T vs. N')
plt.xlim((3.4,5.0))
plt.ylim((-0.5,2.5))
plt.grid(True)
plt.savefig('./log10N_log10T_plot.png',dpi=200)
```

To perform linear fit of $log(T)$ vs. $log(N)$, i.e., $log(T) = \alpha log(N) + \beta$.

It is easy to know that

$$\theta = (X^T X)^{-1} X^T \vec{y}, \tag{1}$$

where $\vec{y} = [log(T_1), \cdots, log(T_k)]^T$, $\theta = [\alpha, \beta]^T$,

$$X = \begin{bmatrix} log(N_1) & 1 \\ log(N_2) & 1 \\ \vdots & \vdots \\ log(N_k) & 1 \end{bmatrix}. \tag{2}$$

Use Python to perform this algorithm:

```
X = np.hstack((log10N.reshape(len(log10N), 1), np.ones(shape=
(len(log10N),1))))
y = log10T.reshape(len(log10T), 1)
pinv = np.linalg.pinv(np.matmul(X.T, X))
theta = np.matmul(np.matmul(pinv, X.T), y)
print('alpha = %.4f, beta = %.4f' % (theta[0], theta[1]))
print('That is\n logT = %.4f * logN %.4f' % (theta[0], theta[1]))
```

The result is $\alpha = 1.9506$

I-2. Theoretical Flop/s Performance

For each clock cycle, each core performs $4 \times 2 = 8 flop$, so

the theoretical peak performance of your computer is
$4 \times 3 \times 10^9 \times 8 flop = 96 \times 10^9 \, flop/s = 96 Gflop/s$.

Part II

File **global_driver.c**:

```
#include "mpi.h"
#include <stdio.h>
int nprocs;  /* Number of processors */
int myid;    /* My rank */

double global_sum(double partial) {
    /* Implement your own global summation here */
    double mydone, hisdone;
    int bitvalue, partner;
    MPI_Status status;
    mydone = partial;

    for(bitvalue=1;bitvalue<nprocs;bitvalue*=2){
        partner = myid ^ bitvalue;
        MPI_Send(&mydone, 1, MPI_DOUBLE,partner, bitvalue, MPI_COMM_WORLD);
 // bitvalue is treated as communication label here
        MPI_Recv(&hisdone, 1, MPI_DOUBLE,partner, bitvalue, MPI_COMM_WORLD,
&status);
        mydone += hisdone;
    }
    return mydone;
}

int main(int argc, char *argv[]) {
    double partial, sum, avg;
```

```
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &myid);
    MPI_Comm_size(MPI_COMM_WORLD, &nprocs);

    partial = (double) myid;
    printf("Rank %d has %le\n", myid, partial);
    sum = global_sum(partial);
    if (myid == 0) {
        avg = sum/nprocs;
        printf("Global average = %le\n", avg);
    }
    MPI_Finalize();
    return 0;
}
```

File **global.sl**:

```
#!/bin/bash
#SBATCH --ntasks-per-node=4
#SBATCH --nodes=2
#SBATCH --time=00:00:59
#SBATCH --output=global.out
#SBATCH -A lc_an2
WORK_HOME=/home/rcf-proj/an2/Your_ID
cd $WORK_HOME
srun -n $SLURM_NTASKS --mpi=pmi2 ./global
srun -n              4 --mpi=pmi2 ./global
```

Output result **global.out**:

```
/var/spool/slurm/slurmd/spool/job1474148/slurm_script: line 8: cd: /home/rcf-
proj/an2/Your_ID: No such file or directory
----------------------------------------
Begin SLURM Prolog Fri 07 Sep 2018 04:21:29 PM PDT
Job ID:        1474148
Username:      liangsiq
Accountname:   lc_an2
Name:          global.sl
Partition:     quick
Nodes:         hpc[4465,4467]
TasksPerNode:  4(x2)
CPUSPerTask:   Default[1]
TMPDIR:        /tmp/1474148.quick
SCRATCHDIR:    /staging/scratch/1474148
Cluster:       uschpc
HSDA Account:  false
End SLURM Prolog
----------------------------------------
Node 4 has 4.000000e+00
Node 0 has 0.000000e+00
Global average = 3.500000e+00
Node 2 has 2.000000e+00
Node 1 has 1.000000e+00
Node 3 has 3.000000e+00
Node 6 has 6.000000e+00
Node 5 has 5.000000e+00
Node 7 has 7.000000e+00
Node 1 has 1.000000e+00
```

```
Node 2 has 2.000000e+00
Node 0 has 0.000000e+00
Global average = 1.500000e+00
Node 3 has 3.000000e+00
```