Homework 1

Chenyue Wu

March 9, 2015

1 Problem1

1.1 Test the program for a large N

On my local machine:

```
$ mpirun -np 4 ./int_ring 10000000
After 10000000 loops on 4 processes, the integer grows to 60000000.
Total communication: 40000000.
Time elapsed: 20.095609 seconds.
Latency: 0.000001 seconds.
```

On remote machine:

```
$ mpirun -np 4 -hosts crunchy1,crunchy3 ./int_ring 10000
After 10000 loops on 4 processes, the integer grows to 60000.
Total communication: 40000.
Time elapsed: 2.303186 seconds.
Latency: 0.000058 seconds.
```

1.2 Array of 2MB

On my local machine

```
mpirun -np 4 ./array_ring.o 1000
Total communication: 4000.
Time elapsed: 2.633973 seconds.
Latency: 0.000658 seconds.
Bandwidth 3.037236GB/s
```

2 Problem2

2.1 Result is independent of p

I save u to different txt files and use 'cmp' to compare the files. It shows that result is independent of p.

2.2 Strong scaling

It is not strong scaling.

```
$ mpirun -np 2 ./jacobi-mpi 100000 10
Results saved in vec100000loop10np2.txt.
Time elapsed: 0.058185 seconds.
$ mpirun -np 4 ./jacobi-mpi 100000 10
Results saved in vec100000loop10np4.txt.
Time elapsed: 0.085044 seconds.
$ mpirun -np 8 ./jacobi-mpi 100000 10
Results saved in vec100000loop10np8.txt.
Time elapsed: 0.484799 seconds.
```

2.3 Parallel version of the Gauss-Seidel smoother

Parallel version of the Gauss-Seidel smoother is more difficult because u needs to be sequentially updated in each step.