CSCI596 Assignment 1—Complexity, Flop/s and Message Passing Interface—Answer

Part I. Computational Complexity and Flop/s Performance

I-1. Measuring Computational Complexity

Figure 1 is a log-log plot of the running time T per MD step of the `md.c` program *vs.* the number of atoms N on a MacBook Pro laptop with 2.7 GHz Core i7 processor. The slope of the linear fit, $\log T = \alpha \log N + \beta$ (α = 1.9506) gives the power of N to which T is proportional. The measured $N^{1.95}$ is close to the expected $O(N^2)$ scaling due to the doubly nested loops in function `ComputeAccel()`.

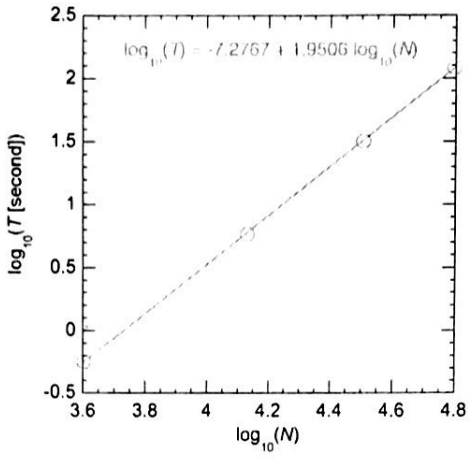

Fig. 1: Log-log plot of running time *vs.* number of atoms for md.c on 2.7 GHz Intel Core i7 processor.

I-2. Theoretical Flop/s Performance

The theoretical peak performance is:
3.0 (GHz) × 2 (operations) × 4 (double-precision numbers) × 4 (cores) = **96 (Gflop/s)**.

Part II. Implementing Your Own Global Summation with Message Passing Interface

Program: `global_avg.c`

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

double global_sum(double partial) {
  MPI_Status status;
  int bitvalue, partner;
  double mydone, hisdone;

  mydone = partial;
  for (bitvalue=1; bitvalue<nprocs; bitvalue *= 2) {
    partner = myid ^ bitvalue; /* XOR flips the l-th bit */
    MPI_Send(&mydone, 1, MPI_DOUBLE, partner, bitvalue, MPI_COMM_WORLD);
    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;
  double cpu1,cpu2;

  MPI_Init(&argc, &argv);
```

```
  MPI_Comm_rank(MPI_COMM_WORLD, &myid);
  MPI_Comm_size(MPI_COMM_WORLD, &nprocs);
  partial = (double) myid;
  printf("Node %d has %le\n", myid, partial);
  cpu1 = MPI_Wtime();
  sum = global_sum(partial);
  cpu2 = MPI_Wtime();
  if (myid == 0) {
    avg = sum/nprocs;
    printf("Global average = %le\n", avg);
    printf("Execution time (s) = %le\n", cpu2-cpu1);
  }
  MPI_Finalize();
  return 0;
}
```

Slurm Script

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

Sample Output

```
[anakano@hpc-login3]$ more global_avg.out
----------------------------------------
Begin SLURM Prolog Sun 02 Sep 2018 01:50:36 PM PDT
Job ID:         1443047
Username:       anakano
Accountname:    lc_an2
Name:           global_avg.sl
Partition:      quick
Nodes:          hpc[1120-1121]
TasksPerNode:   4(x2)
CPUSPerTask:    Default[1]
TMPDIR:         /tmp/1443047.quick
SCRATCHDIR:     /staging/scratch/1443047
Cluster:        uschpc
HSDA Account:   false
End SLURM Prolog
----------------------------------------
...
Node 0 has 0.000000e+00
Node 1 has 1.000000e+00
Node 2 has 2.000000e+00
Node 3 has 3.000000e+00
Node 5 has 5.000000e+00
Node 6 has 6.000000e+00
Node 4 has 4.000000e+00
Node 7 has 7.000000e+00
Global average = 3.500000e+00
Execution time (s) = 6.477118e-03

Node 0 has 0.000000e+00
Node 1 has 1.000000e+00
Node 2 has 2.000000e+00
Node 3 has 3.000000e+00
Global average = 1.500000e+00
Execution time (s) = 6.828308e-04
```