PARALLEL COMPUTING

ASSIGNMENT 1

SOLVING LOWER TRIANGULAR SYSTEM OF LINEAR EQUATIONS

STEPS TO RUN THE SUBMISSION

(Instructions to run the OpenMP version.)

1. Compile the file using the following command:
   ```
   icc -O -openmp ./openMP_ver.cu
   ```

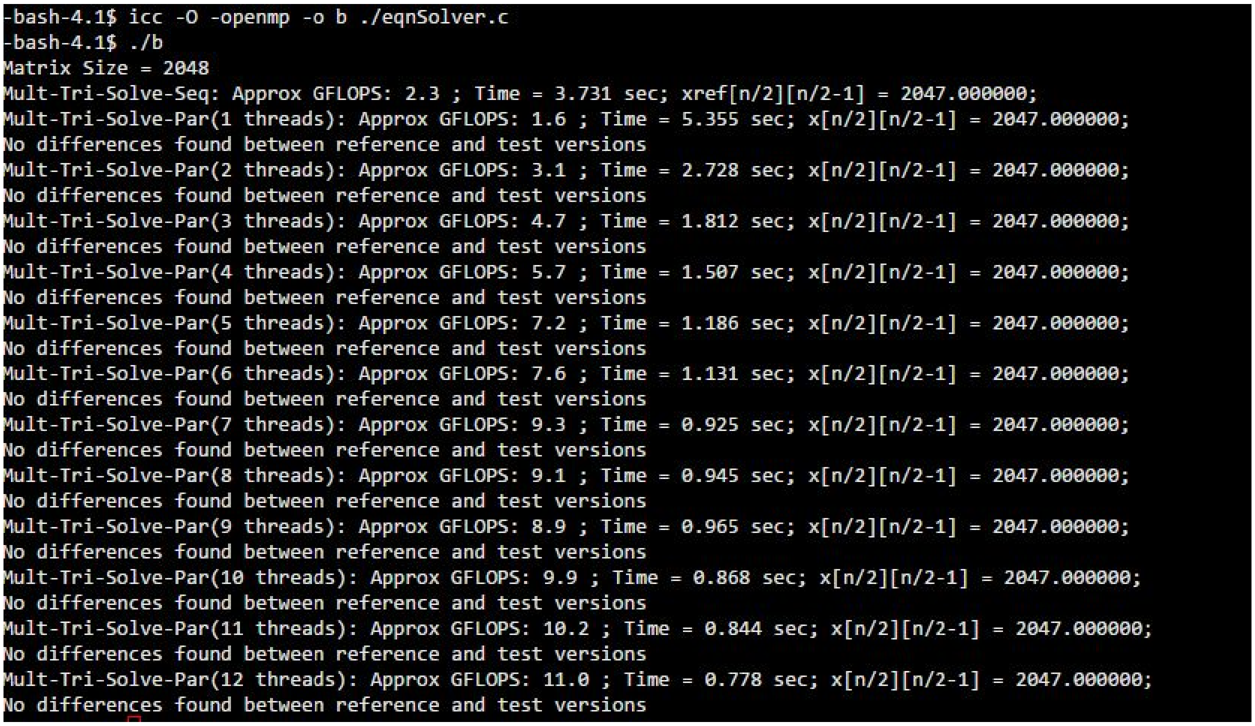
2. Run the output .
   ```
   ./a.out
   ```

(Instructions to run the cuda version.)

2. Compile the file using the following command:
   ```
   nvcc -O -o cuda_1 ./cuda_1.cu
   ```

2. Run the output .
   ```
   ./cuda_1
   ```

OpenMP version



```
-bash-4.1$ icc -O -openmp -o b ./eqnSolver.c
-bash-4.1$ ./b
Matrix Size = 2048
Mult-Tri-Solve-Seq: Approx GFLOPS: 2.3 ; Time = 3.731 sec; xref[n/2][n/2-1] = 2047.000000;
Mult-Tri-Solve-Par(1 threads): Approx GFLOPS: 1.6 ; Time = 5.355 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
Mult-Tri-Solve-Par(2 threads): Approx GFLOPS: 3.1 ; Time = 2.728 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
Mult-Tri-Solve-Par(3 threads): Approx GFLOPS: 4.7 ; Time = 1.812 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
Mult-Tri-Solve-Par(4 threads): Approx GFLOPS: 5.7 ; Time = 1.507 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
Mult-Tri-Solve-Par(5 threads): Approx GFLOPS: 7.2 ; Time = 1.186 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
Mult-Tri-Solve-Par(6 threads): Approx GFLOPS: 7.6 ; Time = 1.131 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
Mult-Tri-Solve-Par(7 threads): Approx GFLOPS: 9.3 ; Time = 0.925 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
Mult-Tri-Solve-Par(8 threads): Approx GFLOPS: 9.1 ; Time = 0.945 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
Mult-Tri-Solve-Par(9 threads): Approx GFLOPS: 8.9 ; Time = 0.965 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
Mult-Tri-Solve-Par(10 threads): Approx GFLOPS: 9.9 ; Time = 0.868 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
Mult-Tri-Solve-Par(11 threads): Approx GFLOPS: 10.2 ; Time = 0.844 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
Mult-Tri-Solve-Par(12 threads): Approx GFLOPS: 11.0 ; Time = 0.778 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
```

OBSERVATIONS:

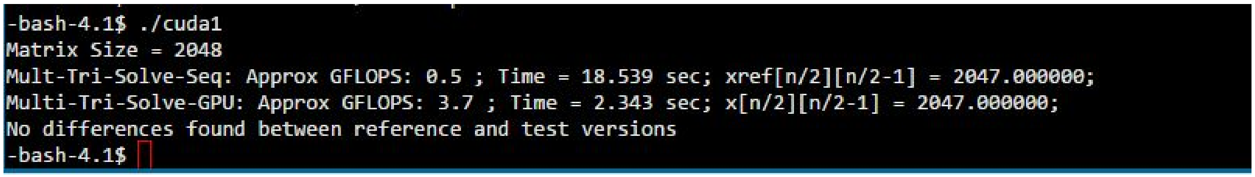
1. Requested number of threads is being allocated by the openMP compiler.
2. The serial version is taking around 3.
3. The performance is improving for increase in the number of threads.
4. The performance is peaking around 11.0 GFLOPS.
5. The increase in performance is not entirely non-decreasing. we observe a slight reduce in performance when we run the algorithm with 9 threads.

CUDA Version

The dimensions of the grid and block were tweaked in order to study the performance.

OBSERVATIONS:

1. The serial version was taking around 18.54s with performance of around 0.5 GFLOPs.
2. For a grid dimension of <32,32> and block dimension of <2,2> a peak performance of 3.7 GFLOPs.



```
-bash-4.1$ ./cuda1
Matrix Size = 2048
Mult-Tri-Solve-Seq: Approx GFLOPS: 0.5 ; Time = 18.539 sec; xref[n/2][n/2-1] = 2047.000000;
Multi-Tri-Solve-GPU: Approx GFLOPS: 3.7 ; Time = 2.343 sec; x[n/2][n/2-1] = 2047.000000;
No differences found between reference and test versions
-bash-4.1$
```