Alireza Safdari Ghandhari　　　　Homework4　　　　11/15/2020
UIN: 728009403　　　　CSCE-735

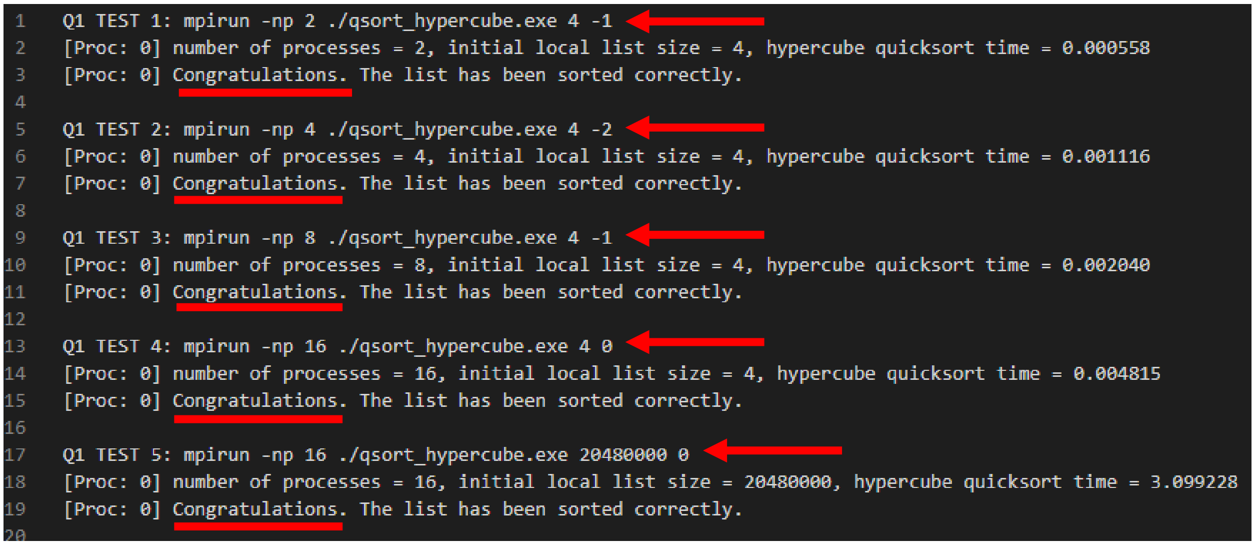
1. (80 points) Complete the MPI-based code provided in qsort_hypercube.cpp to implement the parallel quicksort algorithm for a d-dimensional hypercube with $p=2^d$ processors. 60 points will be awarded if the code compiles and executes the following command successfully.

```
mpirun -np 2 ./qsort_hypercube.exe 4 -1
```

5 points will be awarded for each of the following tests that are executed successfully.

```
mpirun -np 4 ./qsort_hypercube.exe 4 -2
mpirun -np 8 ./qsort_hypercube.exe 4 -1
mpirun -np 16 ./qsort_hypercube.exe 4 0
mpirun -np 16 ./qsort_hypercube.exe 20480000 0
```

My code correctly runs all the cases above, here is the screenshot from output generated after running the job file containing all of them. On change that I have made to my code which is not within MPI lines but does not affect the printed result or the algorithm is returning a bool from HyperCube_QuickSort. This was very beneficial for debugging because I could return at specific points without running the whole function. I decided not to remove this in my final submission because some of debugging code is commented but could be used for printing more information. Also please take note that I have programmed in Visual Studio and it has reformatted the file.



```
Q1 TEST 1: mpirun -np 2 ./qsort_hypercube.exe 4 -1
[Proc: 0] number of processes = 2, initial local list size = 4, hypercube quicksort time = 0.000558
[Proc: 0] Congratulations. The list has been sorted correctly.

Q1 TEST 2: mpirun -np 4 ./qsort_hypercube.exe 4 -2
[Proc: 0] number of processes = 4, initial local list size = 4, hypercube quicksort time = 0.001116
[Proc: 0] Congratulations. The list has been sorted correctly.

Q1 TEST 3: mpirun -np 8 ./qsort_hypercube.exe 4 -1
[Proc: 0] number of processes = 8, initial local list size = 4, hypercube quicksort time = 0.002040
[Proc: 0] Congratulations. The list has been sorted correctly.

Q1 TEST 4: mpirun -np 16 ./qsort_hypercube.exe 4 0
[Proc: 0] number of processes = 16, initial local list size = 4, hypercube quicksort time = 0.004815
[Proc: 0] Congratulations. The list has been sorted correctly.

Q1 TEST 5: mpirun -np 16 ./qsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube quicksort time = 3.099228
[Proc: 0] Congratulations. The list has been sorted correctly.
```

2. (5 points) Weak Scalability Study: Run your code to sort a distributed list of size np where n is the size of the local list on each process and p is the number of processes. For your experiments, use n=20,480,000 and p = 1, 2, 4, 8, 16, 32, and 64. Set type=0. Plot the execution time, speedup, and efficiency of your code as a function of p. Use logarithmic scale for the x-axis.

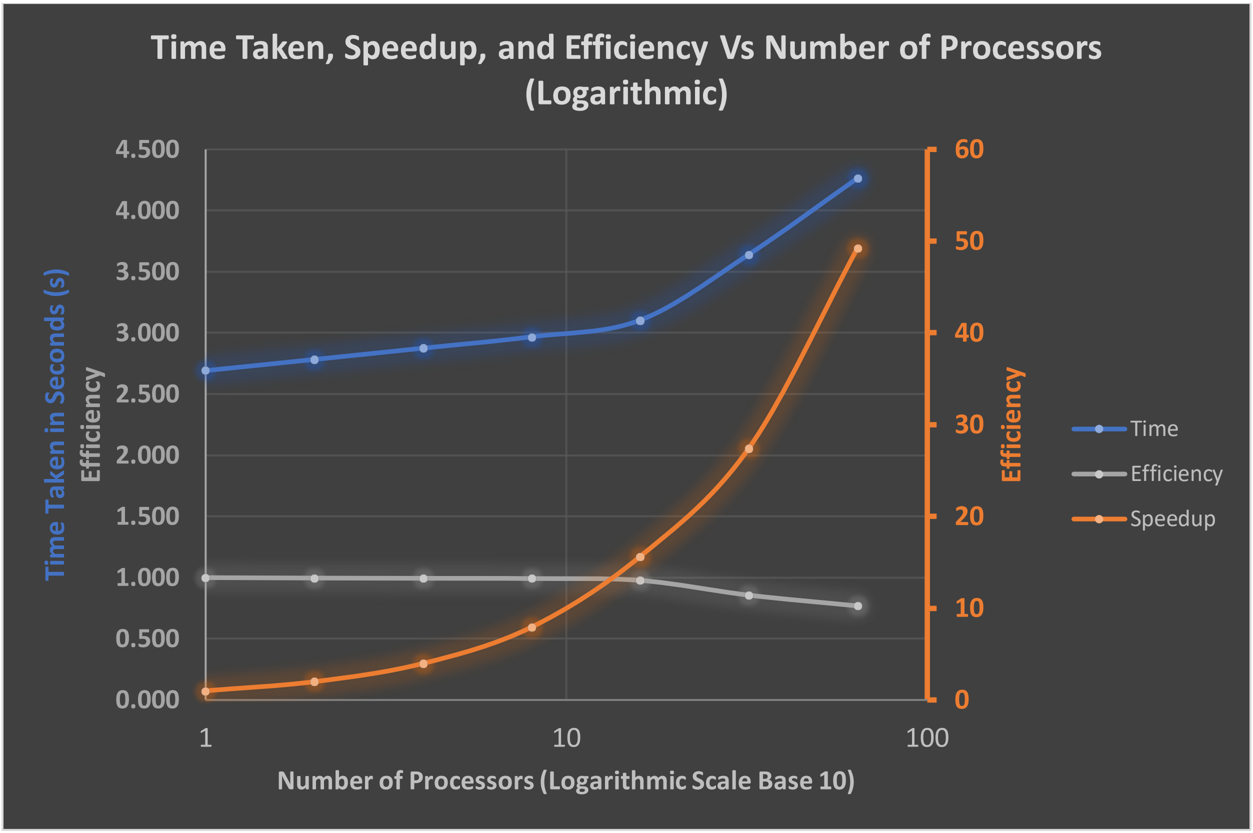
In this table each processor is doing 24,480,000.

# Processes	T1 (s)	T2 (s)	T3 (s)	T4 (s)	T5 (s)	AVG Time (s)	Speedup	Efficiency
1	2.685	2.689	2.682	2.684	2.719	2.692	1	1.00
2	2.775	2.795	2.791	2.761	2.793	2.783	1.93	0.97
4	2.877	2.879	2.872	2.886	2.862	2.875	3.74	0.94
8	2.983	2.971	2.955	2.965	2.968	2.968	7.25	0.91
16	3.113	3.104	3.104	3.102	3.098	3.104	13.87	0.87
32	3.633	3.637	3.649	3.651	3.635	3.641	23.66	0.74
64	4.288	4.257	4.244	4.290	4.240	4.264	40.41	0.63

In this table a single processor is doing the total work from multiprocessor above.

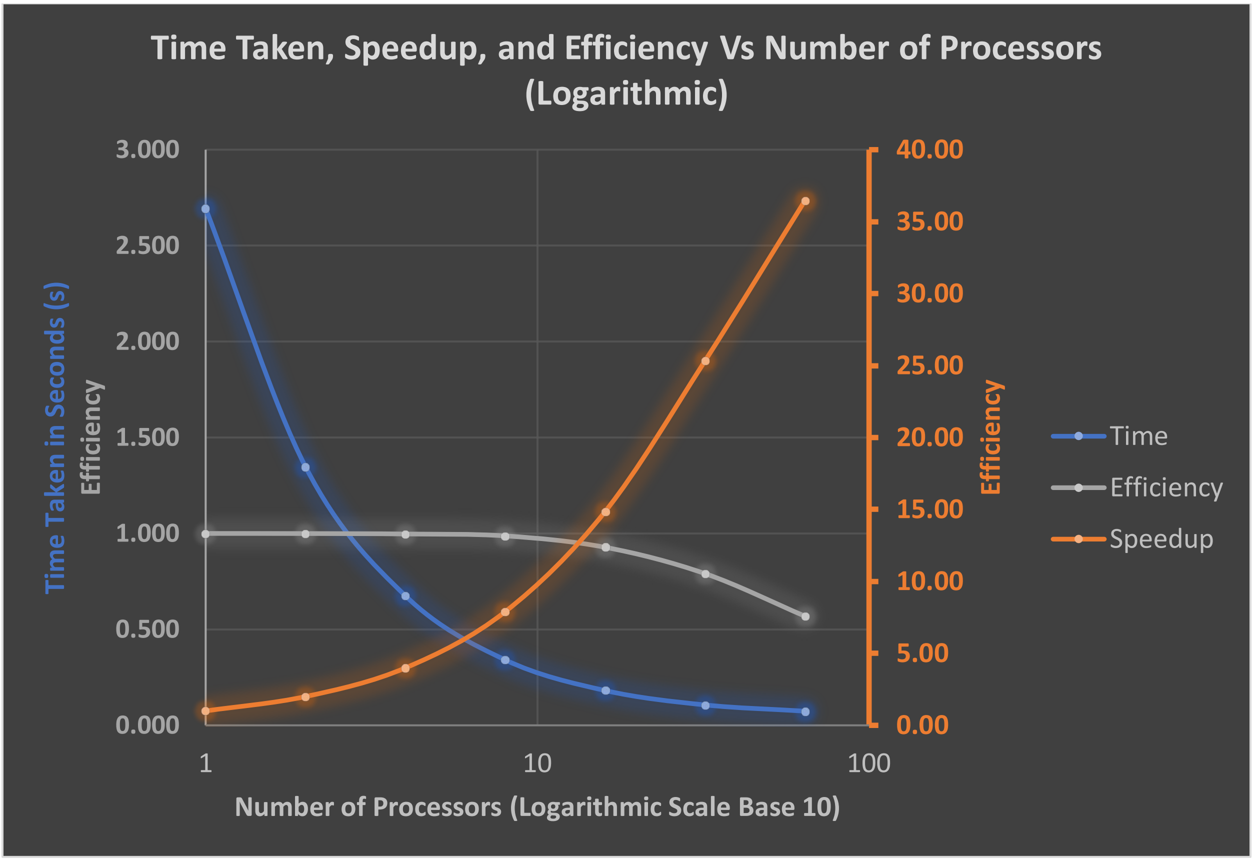
T1 (s)	T2 (s)	T3 (s)	T4 (s)	T5 (s)	AVG Time (s)	Total Points
5.565	5.544	5.549	5.538	5.537	5.547	40960000
11.429	11.433	11.432	11.428	11.467	11.438	81920000
23.587	23.539	23.562	23.565	23.566	23.564	163840000
48.674	48.564	48.467	48.490	48.479	48.535	327680000
100.001	99.704	99.787	99.817	99.818	99.825	655360000
205.087	205.284	216.535	213.104	209.193	209.841	1310720000

To be able to run a single processor versions, I had to change “MPI_Init(&argc,&argv);” to “MPI_Init(NULL, NULL);”


Time Taken, Speedup, and Efficiency Vs Number of Processors (Logarithmic)
Time Taken in Seconds (s)
Efficiency
4.500
4.000
3.500
3.000
2.500
2.000
1.500
1.000
0.500
0.000
60
50
40
30
20
10
0
Efficiency
1
10
100
Number of Processors (Logarithmic Scale Base 10)
Time
Efficiency
Speedup

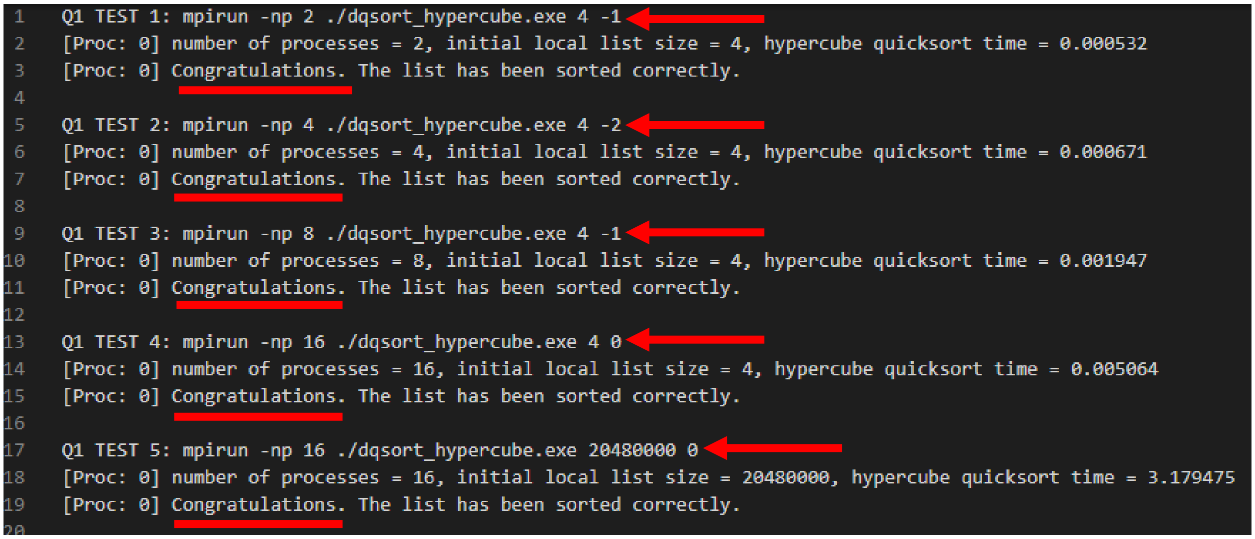
3. (5 points) Strong Scalability Study: Now run your code with n=20,480,000/p where p = 1, 2, 4, 8, 16, 32, and 64. Set type=0. Plot the execution time, speedup, and efficiency of your code as a function of p. Use logarithmic scale for the x-axis.

# Processes	T1 (s)	T2 (s)	T3 (s)	T4 (s)	T5 (s)	AVG Time (s)	Speedup	Efficiency
1	2.684	2.686	2.690	2.689	2.721	2.694	1.00	1.00
2	1.336	1.354	1.349	1.343	1.356	1.348	2.00	1.00
4	0.677	0.677	0.678	0.669	0.676	0.675	3.99	1.00
8	0.339	0.341	0.342	0.341	0.341	0.341	7.90	0.99
16	0.181	0.181	0.182	0.181	0.182	0.181	14.85	0.93
32	0.105	0.105	0.110	0.106	0.105	0.106	25.31	0.79
64	0.073	0.071	0.072	0.072	0.081	0.074	36.44	0.57



4. (10 points) Modify the code to sort the list in descending order. Submit the modified code as qsort_hypercube_descending.cpp. 2 points will be awarded for each of the tests in Problem 1 that are executed successfully. (Note that the check_list routine needs to be modified to verify descending order.)

I have named my file “dqsort_hypercube.cpp”.
I have changed “merged_list”, “split_list_index”, “check_list”, “HyperCube_QuickSort”, and “compare_int”.
I have tested the code and it compiles and run correctly. You can see the result of running the tests in Q1, while there were ran in a single job file.



```
Q1 TEST 1: mpirun -np 2 ./dqsort_hypercube.exe 4 -1
[Proc: 0] number of processes = 2, initial local list size = 4, hypercube quicksort time = 0.000532
[Proc: 0] Congratulations. The list has been sorted correctly.

Q1 TEST 2: mpirun -np 4 ./dqsort_hypercube.exe 4 -2
[Proc: 0] number of processes = 4, initial local list size = 4, hypercube quicksort time = 0.000671
[Proc: 0] Congratulations. The list has been sorted correctly.

Q1 TEST 3: mpirun -np 8 ./dqsort_hypercube.exe 4 -1
[Proc: 0] number of processes = 8, initial local list size = 4, hypercube quicksort time = 0.001947
[Proc: 0] Congratulations. The list has been sorted correctly.

Q1 TEST 4: mpirun -np 16 ./dqsort_hypercube.exe 4 0
[Proc: 0] number of processes = 16, initial local list size = 4, hypercube quicksort time = 0.005064
[Proc: 0] Congratulations. The list has been sorted correctly.

Q1 TEST 5: mpirun -np 16 ./dqsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube quicksort time = 3.179475
[Proc: 0] Congratulations. The list has been sorted correctly.
```