

CSCE 735 Spring 2023

HW 4: Parallel Programming with MPI – Hypercube Quicksort

Name: Shweta Sharma

UIN: 433003780

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
```

Ans:

Following are the output in The Grace portal:

```
[shweta_sharma_1999@grace1 HW4-735]$ mpirun -np 2 ./qsort_hypercube.exe 4 -1
[Proc: 0] number of processes = 2, initial local list size = 4, hypercube quicksort time = 0.010986
[Proc: 0] Congratulations. The list has been sorted correctly.
[shweta_sharma_1999@grace1 HW4-735]$ mpirun -np 4 ./qsort_hypercube.exe 4 -2
[Proc: 0] number of processes = 4, initial local list size = 4, hypercube quicksort time = 0.011098
[Proc: 0] Congratulations. The list has been sorted correctly.
[shweta_sharma_1999@grace1 HW4-735]$ mpirun -np 8 ./qsort_hypercube.exe 4 -1
[Proc: 0] number of processes = 8, initial local list size = 4, hypercube quicksort time = 0.014507
[Proc: 0] Congratulations. The list has been sorted correctly.
[shweta_sharma_1999@grace1 HW4-735]$ mpirun -np 16 ./qsort_hypercube.exe 4 0
[Proc: 0] number of processes = 16, initial local list size = 4, hypercube quicksort time = 0.022491
[Proc: 0] Congratulations. The list has been sorted correctly.
[shweta_sharma_1999@grace1 HW4-735]$ mpirun -np 16 ./qsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube quicksort time = 3.381205
[Proc: 0] Congratulations. The list has been sorted correctly.
```

All the runs executed successfully with zero error.

2. (5 points) Weak Scalability Study: Run your code to sort a distributed list of size $n \times p$ 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 $\text{type}=0$. Plot the execution time, speedup, and efficiency of your code as a function of p . Use logarithmic scale for the x -axis.

Note that the size of the list to be sorted is proportional to the number of processes p . In order to get speedup for a specific value of p , you need to determine the execution time to sort a list of size $n \times p$ with one process. As an example, speedup for $p = 4$ is the ratio of execution time for a list of size $81,920,000$ with one process (T_1) to the execution time for a list of size $20,480,000$ with 4 processes (T_4).

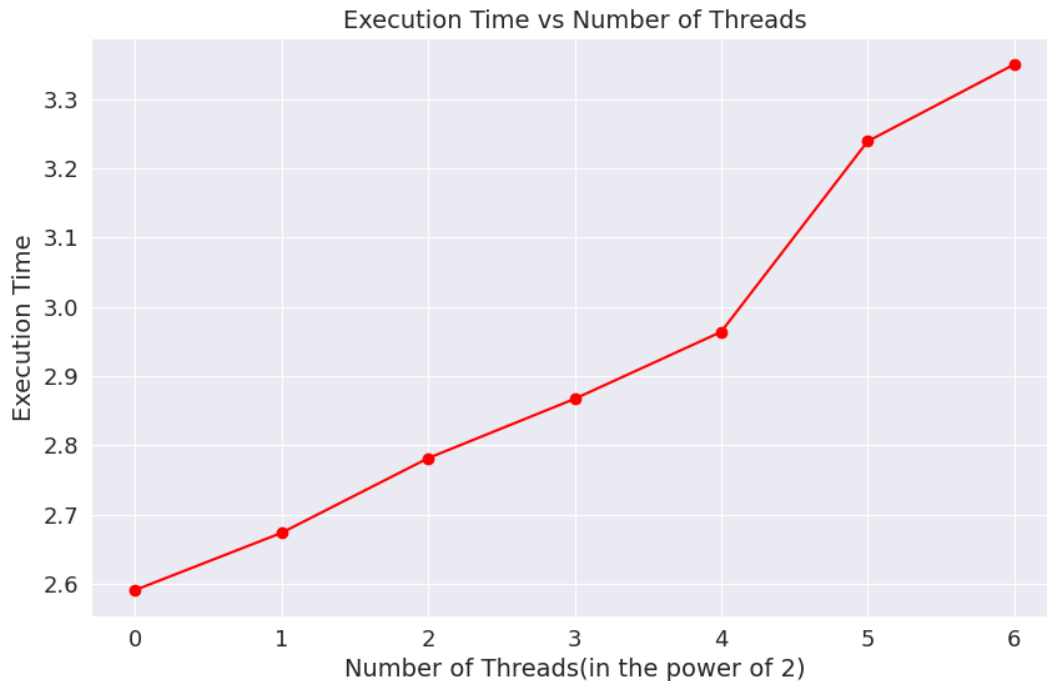
Ans: Following are the output in The Grace portal:

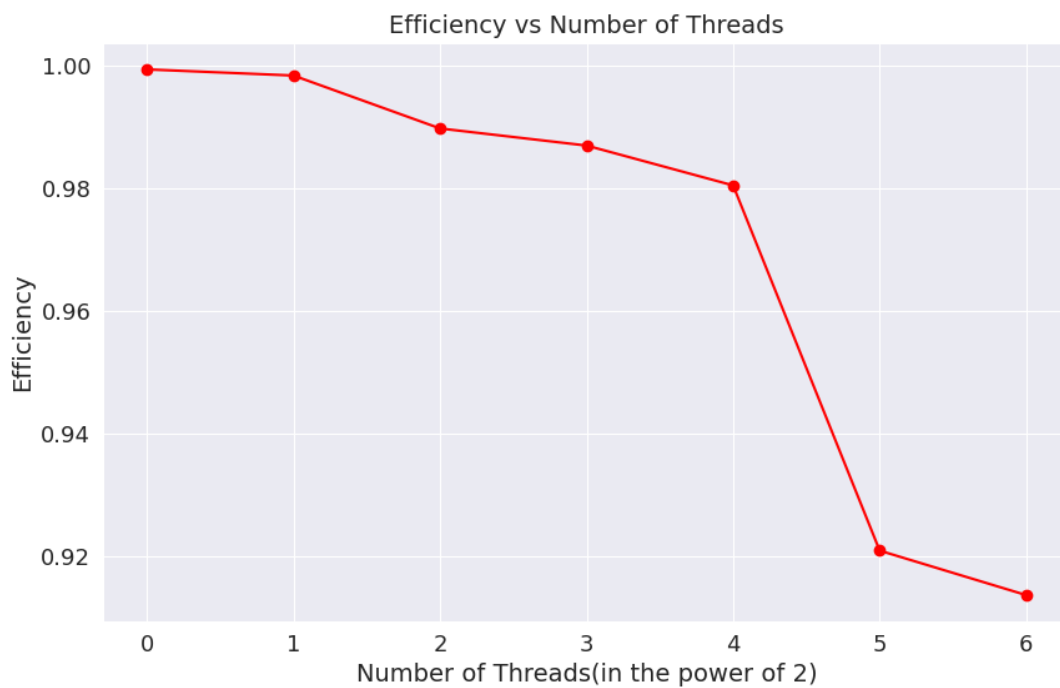
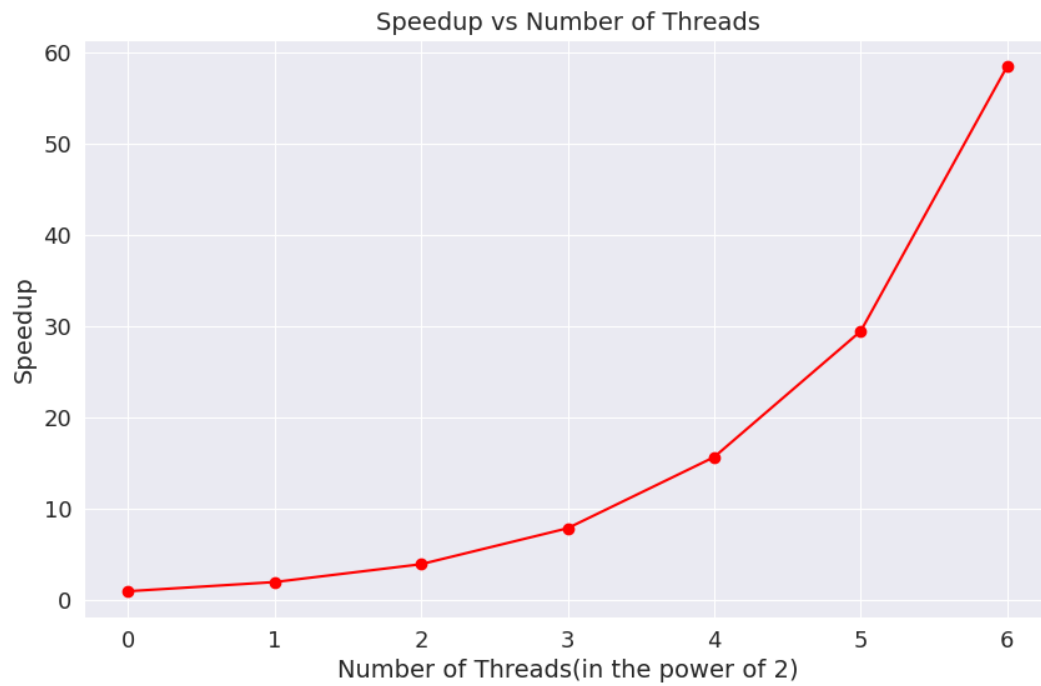
```

[Proc: 0] number of processes = 1, initial local list size = 20480000, hypercube quicksort time = 2.589749
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 40960000, hypercube quicksort time = 5.360384
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 81920000, hypercube quicksort time = 11.012364
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 163840000, hypercube quicksort time = 22.638362
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 327680000, hypercube quicksort time = 46.495310
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 655360000, hypercube quicksort time = 95.438797
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 1310720000, hypercube quicksort time = 195.857297
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 20480000, hypercube quicksort time = 2.591351
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 2, initial local list size = 20480000, hypercube quicksort time = 2.673935
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 4, initial local list size = 20480000, hypercube quicksort time = 2.781617
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 8, initial local list size = 20480000, hypercube quicksort time = 2.867235
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube quicksort time = 2.963906
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 32, initial local list size = 20480000, hypercube quicksort time = 3.238840
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 64, initial local list size = 20480000, hypercube quicksort time = 3.349736
[Proc: 0] Congratulations. The list has been sorted correctly.
output.7917605 (END)

```

The output curves are as follows:





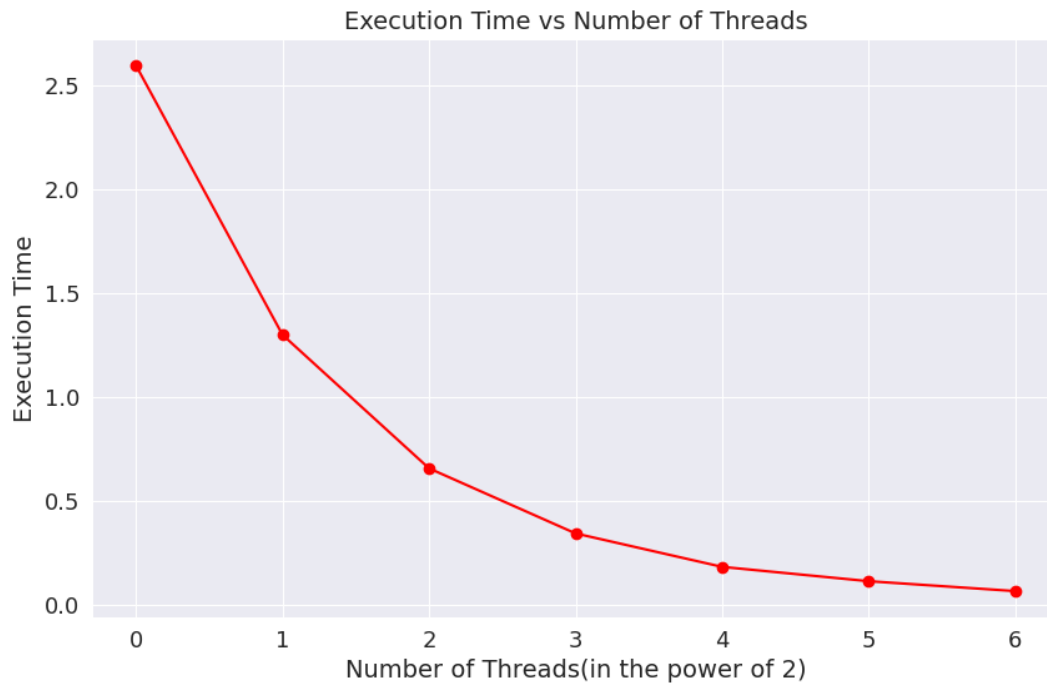
3. (5 points) Strong Scalability Study: Now run your code with $n=20,480,000/p$ where $p = 1, 2, 4, 8, 16, 32$, and 64 . Set $\text{type}=0$. Plot the execution time, speedup, and efficiency of your code as a function of p . Use logarithmic scale for the x-axis.

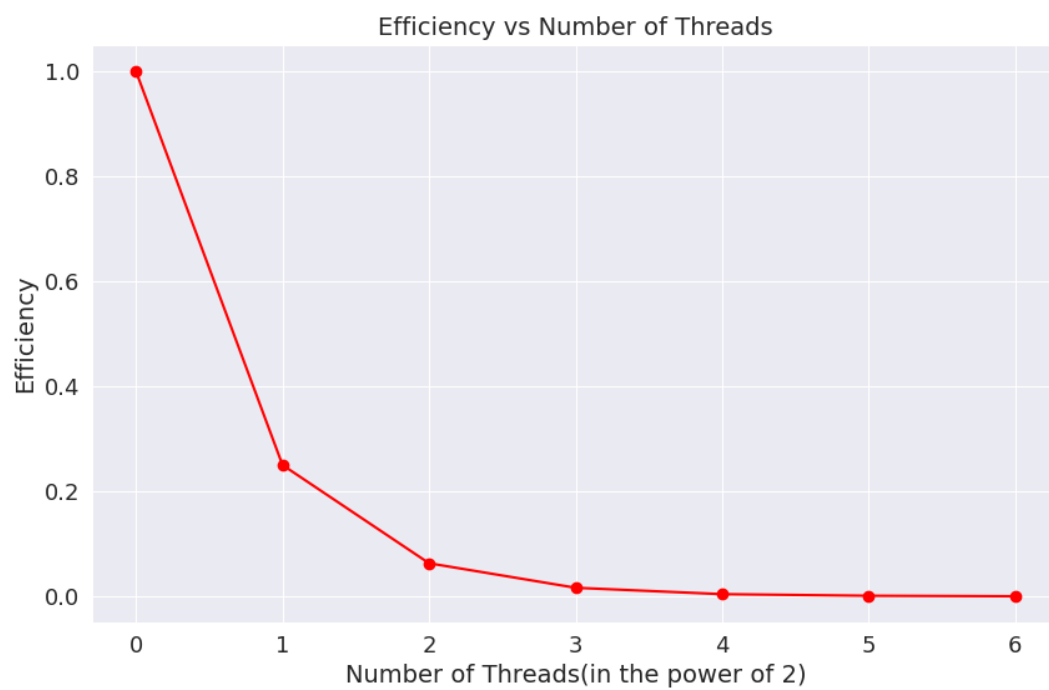
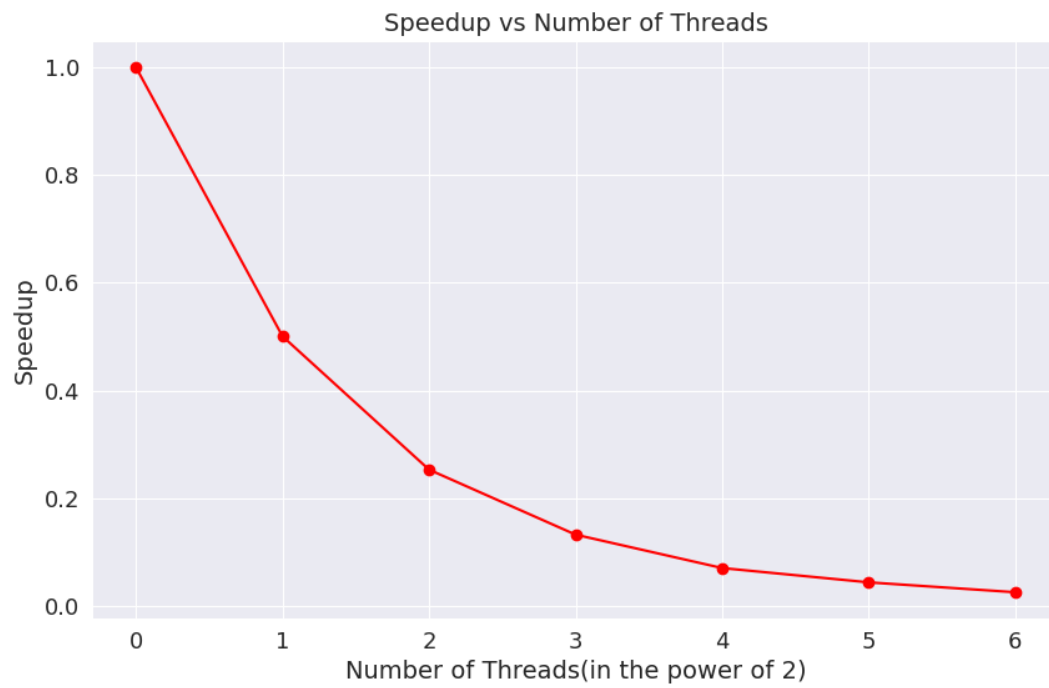
Unlike the weak scalability study, here the size of the list to be sorted remains unchanged at 20,480,000 even as you increase the number of processes. To determine speedup for any p you need to compare the execution time on p processes to the execution time for a list of size 20,480,000 with one process.

Ans: Following are the output in The Grace portal:

```
[Proc: 0] number of processes = 1, initial local list size = 20480000, hypercube quicksort time = 2.593899
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 2, initial local list size = 10240000, hypercube quicksort time = 1.298414
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 4, initial local list size = 5120000, hypercube quicksort time = 0.656137
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 8, initial local list size = 2560000, hypercube quicksort time = 0.343217
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 16, initial local list size = 1280000, hypercube quicksort time = 0.182382
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 32, initial local list size = 640000, hypercube quicksort time = 0.113467
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 64, initial local list size = 320000, hypercube quicksort time = 0.065837
[Proc: 0] Congratulations. The list has been sorted correctly.
output.7917619 (END)
```

The output curves are as follows:





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.)

Ans: Following are the output in The Grace portal:

```
[shweta_sharma_1999@grace1 HW4-735]$ mpirun -np 2 ./qsort_hypercube_descending.exe 4 -1
[Proc: 0] number of processes = 2, initial local list size = 4, hypercube quicksort time = 0.013191
[Proc: 0] Congratulations. The list has been sorted correctly.
[shweta_sharma_1999@grace1 HW4-735]$ mpirun -np 4 ./qsort_hypercube_descending.exe 4 -2
[Proc: 0] number of processes = 4, initial local list size = 4, hypercube quicksort time = 0.014957
[Proc: 0] Congratulations. The list has been sorted correctly.
[shweta_sharma_1999@grace1 HW4-735]$ mpirun -np 8 ./qsort_hypercube_descending.exe 4 -1
[Proc: 0] number of processes = 8, initial local list size = 4, hypercube quicksort time = 0.012877
[Proc: 0] Congratulations. The list has been sorted correctly.
[shweta_sharma_1999@grace1 HW4-735]$ mpirun -np 16 ./qsort_hypercube_descending.exe 4 0
[Proc: 0] number of processes = 16, initial local list size = 4, hypercube quicksort time = 0.008896
[Proc: 0] Congratulations. The list has been sorted correctly.
[shweta_sharma_1999@grace1 HW4-735]$ mpirun -np 16 ./qsort_hypercube_descending.exe 20480000 0
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube quicksort time = 3.146296
[Proc: 0] Congratulations. The list has been sorted correctly.
[shweta_sharma_1999@grace1 HW4-735]$
```