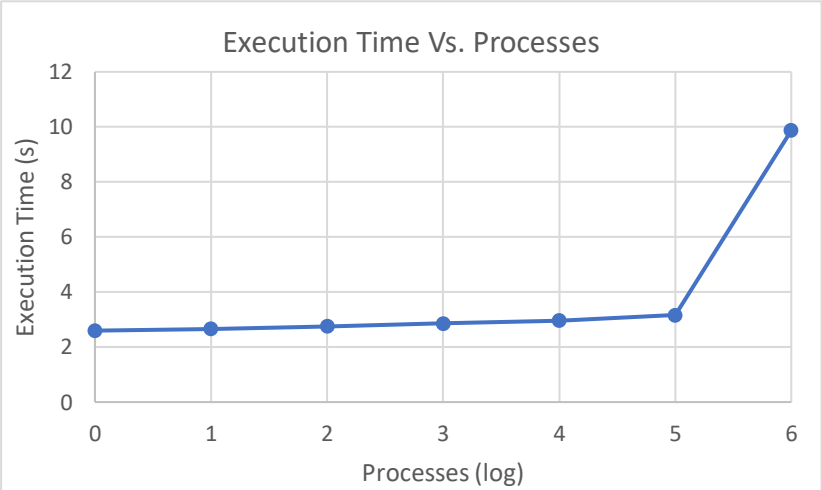
CSCE 735 Parallel Computing – HW4

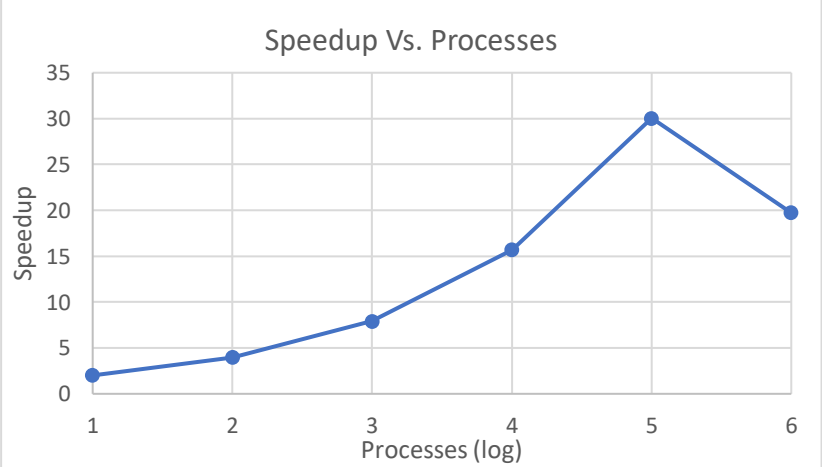
832001192
Chun-Sheng Wu

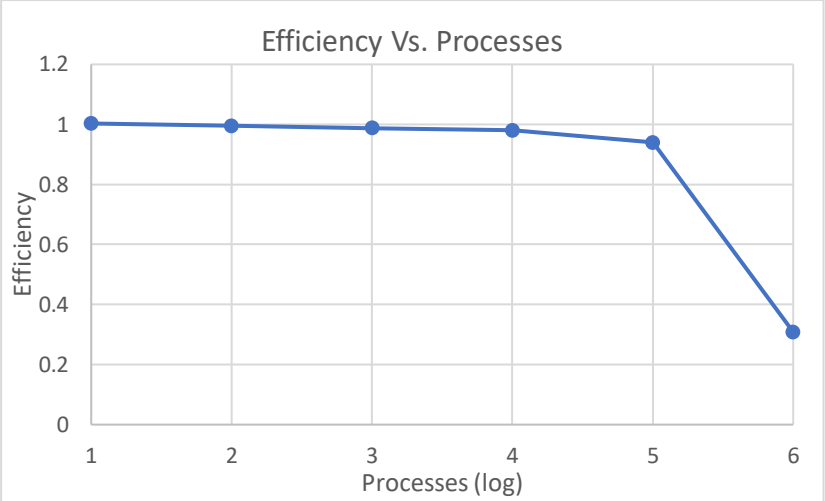
1. Complete modifying qsort_hypercube.cpp and pass the following instructions.

```
[jinsonwu@grace2 HW4]$ mpirun -np 2 ./qsort_hypercube.exe 4 -1
[Proc: 0] number of processes = 2, initial local list size = 4, hypercube quicksort time = 0.004093
[Proc: 0] Congratulations. The list has been sorted correctly.
[jinsonwu@grace2 HW4]$ mpirun -np 4 ./qsort_hypercube.exe 4 -2
[Proc: 0] number of processes = 4, initial local list size = 4, hypercube quicksort time = 0.001620
[Proc: 0] Congratulations. The list has been sorted correctly.
[jinsonwu@grace2 HW4]$ mpirun -np 8 ./qsort_hypercube.exe 4 -1
[Proc: 0] number of processes = 8, initial local list size = 4, hypercube quicksort time = 0.002618
[Proc: 0] Congratulations. The list has been sorted correctly.
[jinsonwu@grace2 HW4]$ mpirun -np 16 ./qsort_hypercube.exe 4 0
[Proc: 0] number of processes = 16, initial local list size = 4, hypercube quicksort time = 0.005544
[Proc: 0] Congratulations. The list has been sorted correctly.
[jinsonwu@grace2 HW4]$ mpirun -np 16 ./qsort_hypercube.exe 20480000 0
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube quicksort time = 2.665893
[Proc: 0] Congratulations. The list has been sorted correctly.
```

2. Weak Scalability Study





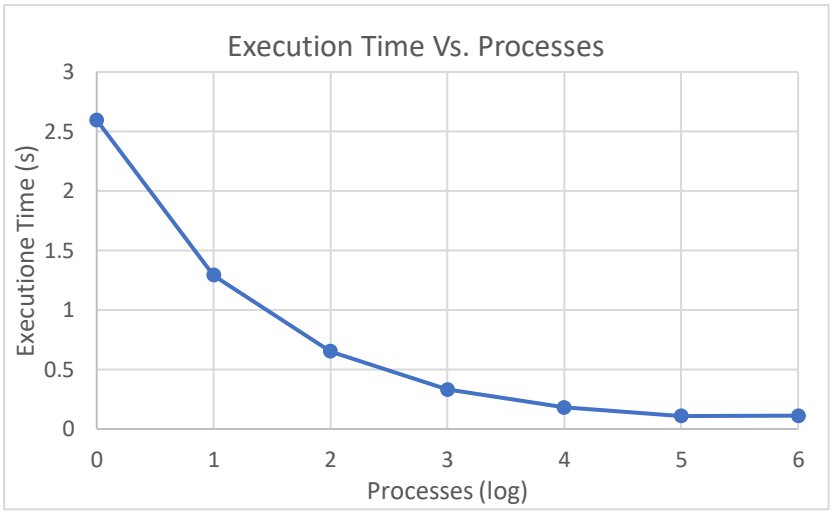


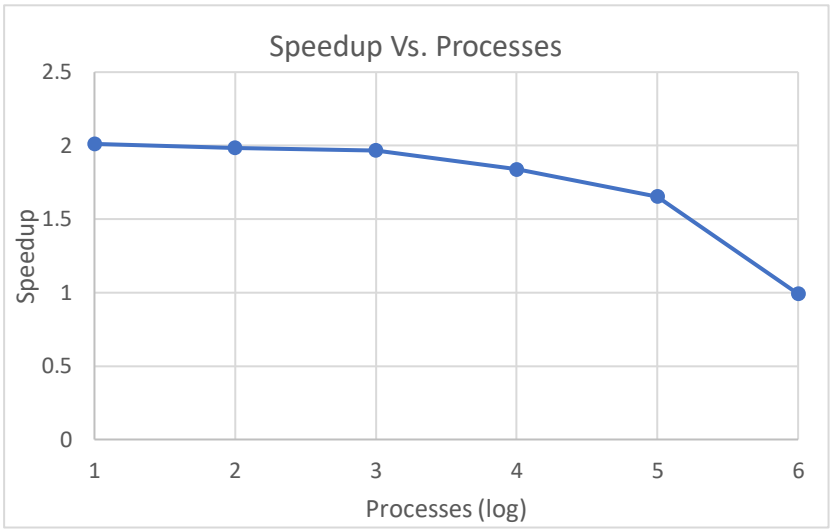
Above are the graphs for execution time, speedup, and efficiency respectively in weak scalability study. The execution time remains similar with processes below 32, and speedup grows nearly linearly accordingly. Efficiency value significantly drops when processes increase to 64. It might be caused by intrinsic hardware resources in Grace. Over-revoking nodes bring about additional modulation and communication, therefore raising the execution time with more nodes than 32.

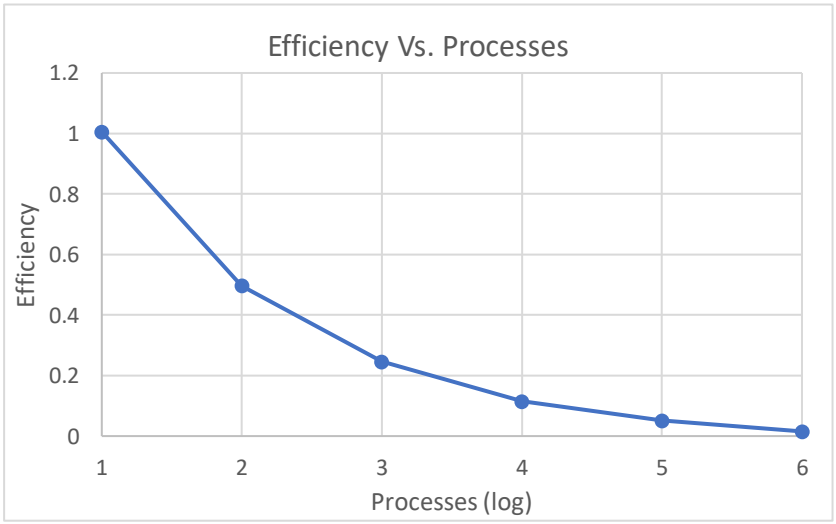
```
Weak Scalability Study
[Proc: 0] number of processes = 1, initial local list size = 20480000, hypercube quicksort time = 2.595876
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 40960000, hypercube quicksort time = 5.338474
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 81920000, hypercube quicksort time = 10.979132
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 163840000, hypercube quicksort time = 22.566414
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 327680000, hypercube quicksort time = 46.422641
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 655360000, hypercube quicksort time = 95.090869
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 1, initial local list size = 1310720000, hypercube quicksort time = 194.984325
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 2, initial local list size = 20480000, hypercube quicksort time = 2.661155
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 4, initial local list size = 20480000, hypercube quicksort time = 2.758070
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 8, initial local list size = 20480000, hypercube quicksort time = 2.855079
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube quicksort time = 2.960000
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 32, initial local list size = 20480000, hypercube quicksort time = 3.164153
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 64, initial local list size = 20480000, hypercube quicksort time = 9.868281
[Proc: 0] Congratulations. The list has been sorted correctly.
```

Experiment result in weak scalability study

3. Strong Scalability Study







In contrast to weak scalability study, performance in all three numeric behaves worse with 4 or more processes in strong scalability study. To handle identical tasks (nxp = 20480000), choose p = 1 or 2 can achieve the best speedup and efficiency in this experiment.

```
Strong Scalability Study
[Proc: 0] number of processes = 2, initial local list size = 10240000, hypercube quicksort time = 1.290415
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 4, initial local list size = 5120000, hypercube quicksort time = 0.650201
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 8, initial local list size = 2560000, hypercube quicksort time = 0.330397
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 16, initial local list size = 1280000, hypercube quicksort time = 0.179691
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 32, initial local list size = 640000, hypercube quicksort time = 0.108702
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 64, initial local list size = 320000, hypercube quicksort time = 0.109700
[Proc: 0] Congratulations. The list has been sorted correctly.
```

Experiment result in strong scalability study

4. Modify the code to have qsort in descending order, pass the commands in 1.

```
[jinsonwu@grace2 HW4]$ mpirun -np 2 ./qsort_hypercube_descending.exe 4 -1
[Proc: 0] number of processes = 2, initial local list size = 4, hypercube quicksort time = 0.004810
[Proc: 0] Congratulations. The list has been sorted correctly.
[jinsonwu@grace2 HW4]$ mpirun -np 4 ./qsort_hypercube_descending.exe 4 -2
[Proc: 0] number of processes = 4, initial local list size = 4, hypercube quicksort time = 0.001894
[Proc: 0] Congratulations. The list has been sorted correctly.
[jinsonwu@grace2 HW4]$ mpirun -np 8 ./qsort_hypercube_descending.exe 4 -1
[Proc: 0] number of processes = 8, initial local list size = 4, hypercube quicksort time = 0.002761
[Proc: 0] Congratulations. The list has been sorted correctly.
[jinsonwu@grace2 HW4]$ mpirun -np 16 ./qsort_hypercube_descending.exe 4 0
[Proc: 0] number of processes = 16, initial local list size = 4, hypercube quicksort time = 0.005425
[Proc: 0] Congratulations. The list has been sorted correctly.
[jinsonwu@grace2 HW4]$ mpirun -np 16 ./qsort_hypercube_descending.exe 20480000 0
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube quicksort time = 2.649583
[Proc: 0] Congratulations. The list has been sorted correctly.
[jinsonwu@grace2 HW4]$
```