CSCE 735 Homework 4

Name: Pranav Anantharam
UIN: 734003886

1. The revised code (MPI-based parallel quicksort for a d-dimensional hypercube) produced the below output for varying number of processes, local list sizes and types (method used to initialize local list):

Spreadsheet Screenshot:

Local List Size	Number of Processes	Local List Initialization Type	Hypercube Quicksort Time (in seconds)
4	2	-1	0.284601
4	4	-2	0.020124
4	8	-1	0.007504
4	16	0	0.010673
20480000	16	0	3.107837

Console Output:

```
[Proc: 0] number of processes = 2, initial local list size = 4, hypercube quicksort time
= 0.284601
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 4, initial local list size = 4, hypercube quicksort time
= 0.020124
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 8, initial local list size = 4, hypercube quicksort time
= 0.007504
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 16, initial local list size = 4, hypercube quicksort time
= 0.010673
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube
quicksort time = 3.107837
[Proc: 0] Congratulations. The list has been sorted correctly.
```

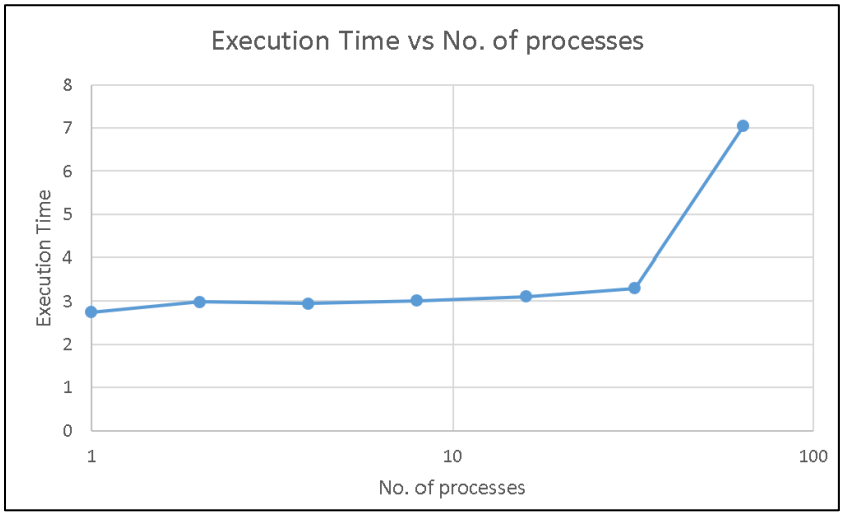
2. Weak Scalability Study:

To conduct the Weak Scalability Study, the code was run with n = 20480000 and p values of 1, 2, 4, 8, 16, 32, and 64 with type set to 0. The following results were obtained:

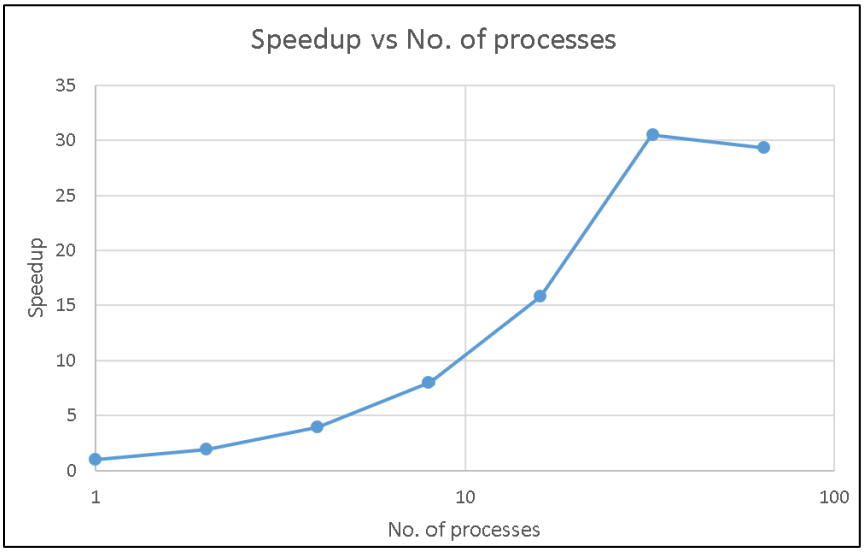
Spreadsheet Calculations:

List Size	Number of Processes	Hypercube Quicksort Time (in seconds)	Hypercube Quicksort Time (in seconds) for 1 processor	Speedup	Efficiency
20480000	1	2.743377	2.743377	1	1
40960000	2	2.982323	5.642125	1.891855778	0.945927889
81920000	4	2.943107	11.639293	3.954763792	0.988690948
163840000	8	3.003136	23.854594	7.943228012	0.992903502
327680000	16	3.10917	49.15763	15.81053143	0.988158214
655360000	32	3.301772	100.6508	30.48387351	0.952621047
1310720000	64	7.047291	206.807331	29.34564941	0.458525772

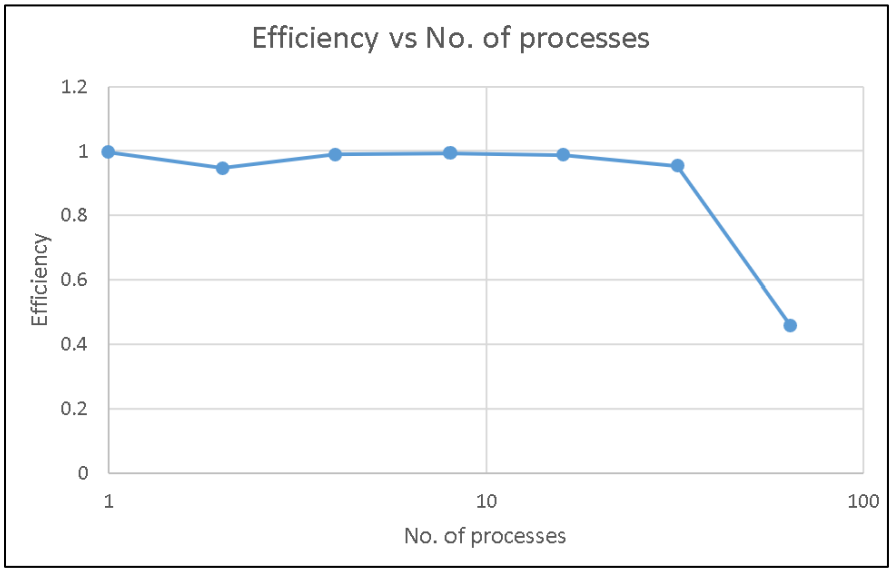
Execution Time vs No. of processes:



Speedup vs No. of processes:



Efficiency vs No. of processes:



---

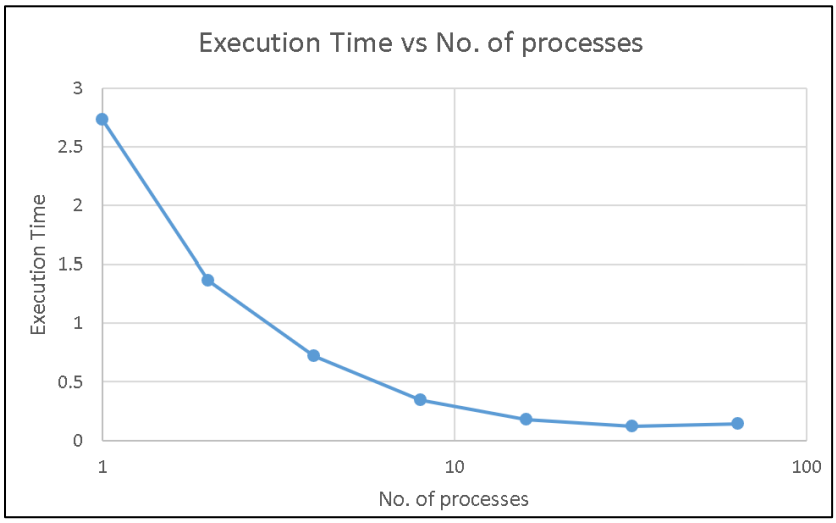
3. Strong Scalability Study:

To conduct the Strong Scalability Study, the code was run with n = 20480000/p and p values of 1, 2, 4, 8, 16, 32, and 64 with type set to 0. The following results were obtained:

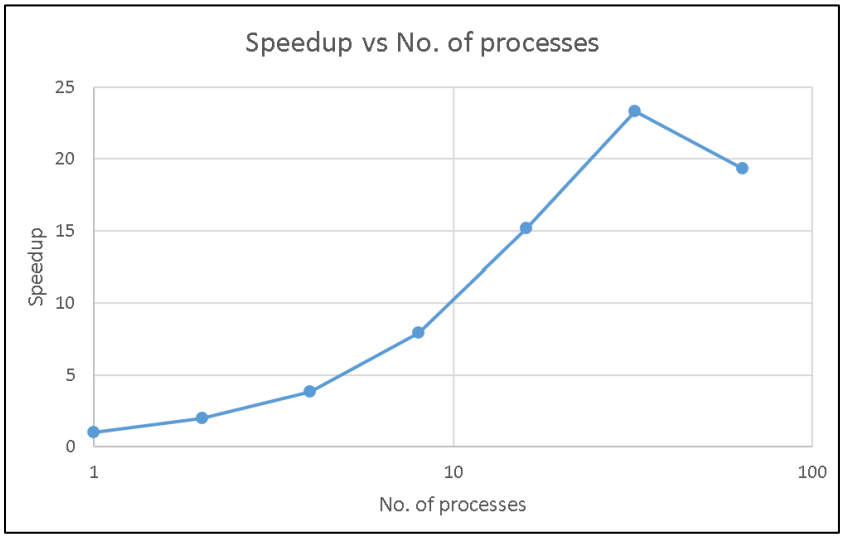
Spreadsheet Calculations:

List Size	Number of Processes	List Size per process	Hypercube Quicksort Time (in seconds)	Hypercube Quicksort Time (in seconds) for 1 processor	Speedup	Efficiency
20480000	1	20480000	2.734435	2.734435	1	1
20480000	2	10240000	1.366632	2.734435	2.000856851	1.000428426
20480000	4	5120000	0.718775	2.734435	3.804298981	0.951074745
20480000	8	2560000	0.344593	2.734435	7.935259857	0.991907482
20480000	16	1280000	0.180239	2.734435	15.17116162	0.948197602
20480000	32	640000	0.117114	2.734435	23.34848951	0.729640297
20480000	64	320000	0.141592	2.734435	19.31207272	0.301751136

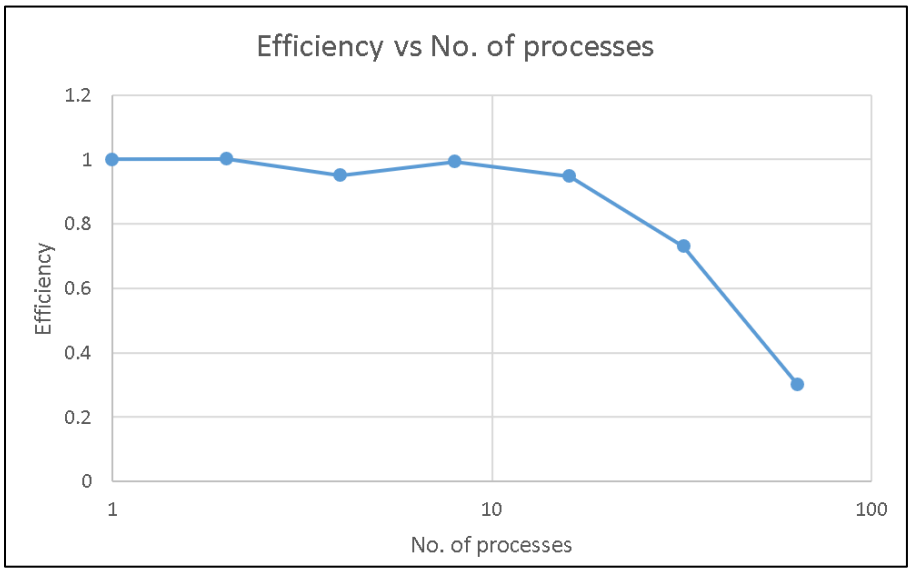
Execution Time vs No. of processes:



Speedup vs No. of processes:



Efficiency vs No. of processes:



---

4. The modified code (MPI-based parallel quicksort for a d-dimensional hypercube) to sort the list in **descending order** produced the below output for varying number of processes, local list sizes and types (method used to initialize local list):

Spreadsheet Screenshot:

Local List Size	Number of Processes	Local List Initialization Type	Hypercube Quicksort Time (in seconds)
4	2	-1	0.001648
4	4	-2	0.002532
4	8	-1	0.007322
4	16	0	0.010637
20480000	16	0	3.122685

Console Output:

```
[Proc: 0] number of processes = 2, initial local list size = 4, hypercube quicksort time
= 0.001648
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 4, initial local list size = 4, hypercube quicksort time
= 0.002532
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 8, initial local list size = 4, hypercube quicksort time
= 0.007322
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 16, initial local list size = 4, hypercube quicksort time
= 0.010637
[Proc: 0] Congratulations. The list has been sorted correctly.
[Proc: 0] number of processes = 16, initial local list size = 20480000, hypercube
quicksort time = 3.122685
[Proc: 0] Congratulations. The list has been sorted correctly.
```
