Cpt S 411 Assignment Cover Sheet

(To be turned in along with each homework and program project submission)

Assignment # 2

For individual assignments:

Student name (Last, First): Cuevas, Jessica

For team projects:

List of all students (Last, First): Cuevas, Jessica

List of collaborative personnel (excluding team participants):
N/A

I[1] certify that I have listed above all the sources that I consulted regarding this assignment, and that I have not received or given any assistance that is contrary to the letter or the spirit of the collaboration guidelines for this assignment. I also certify that I have not referred to online solutions that may be available on the web or sought the help of other students outside the class, in preparing my solution. I attest that the solution is my own and if evidence is found to the contrary, I understand that I will be subject to the academic dishonesty policy as outlined in the course syllabus.

Please print your names.
Jessica Cuevas

Assignment Project Participant(s):
Jessica Cuevas

Today’s Date:
10/09/20

[1] If you worked as a team, then the word “I” includes yourself and your team members.

Programming Project 2 Report

Table and Chart Results:

Table 1. Average Time per. Generation (ms)

Gen = 10	n=4	n=8	n=16	n=32	n=64	n=128	n=256	n=512	n=1024
p=1	0	0	0	0	0	3	12	49	181
p=2	1	0	0	0	0	1	6	25	99
p=4	NA	0	0	0	0	0	3	13	50
p=8	NA	NA	0	0	0	0	1	6	26
p=16	NA	NA	NA	3	4	6	11	21	46
p=32	NA	NA	NA	NA	4	6	10	19	40
p=56	NA	NA	NA	NA	5	7	11	21	41

Table 2. Average Time per. Generation (ms)

Gen = 10000	n=4	n=8	n=16	n=32	n=64	n=128	n=256	n=512
p=1	0.0058	0.0164	0.0591	0.2135	0.8067	3.1357	12.3626	49.072201
p=2	0.0055	0.0135	0.0391	0.1256	0.4413	1.6528	6.3352	24.8475
p=4	NA	0.0137	0.0337	0.0903	0.2751	0.9326	3.4008	12.848
p=8	NA	NA	0.0321	0.0744	0.1955	0.579	1.9162	6.8675
p=16	NA	NA	NA	1.2371	2.2869	4.4686	9.0693	19.433901
p=32	NA	NA	NA	NA	2.369	4.467	8.8023	18.045099
p=56	NA	NA	NA	NA	NA	NA	NA	NA

*Note: Do to overflow of requests in the cluster I was unable to test n=1024. Also p=56 is not divisible by all input sizes thus no data was recorded.

Table 3. Speedup (T(n,1) / T(n,p))

p	n=64	n=128	n=256	n=512
1	1.00	1.00	1.00	1.00
2	1.83	1.90	1.95	1.97
4	2.93	3.36	3.64	3.82
8	4.13	5.42	6.45	7.15
16	0.35	0.70	1.36	2.53
32	0.34	0.70	1.40	2.72

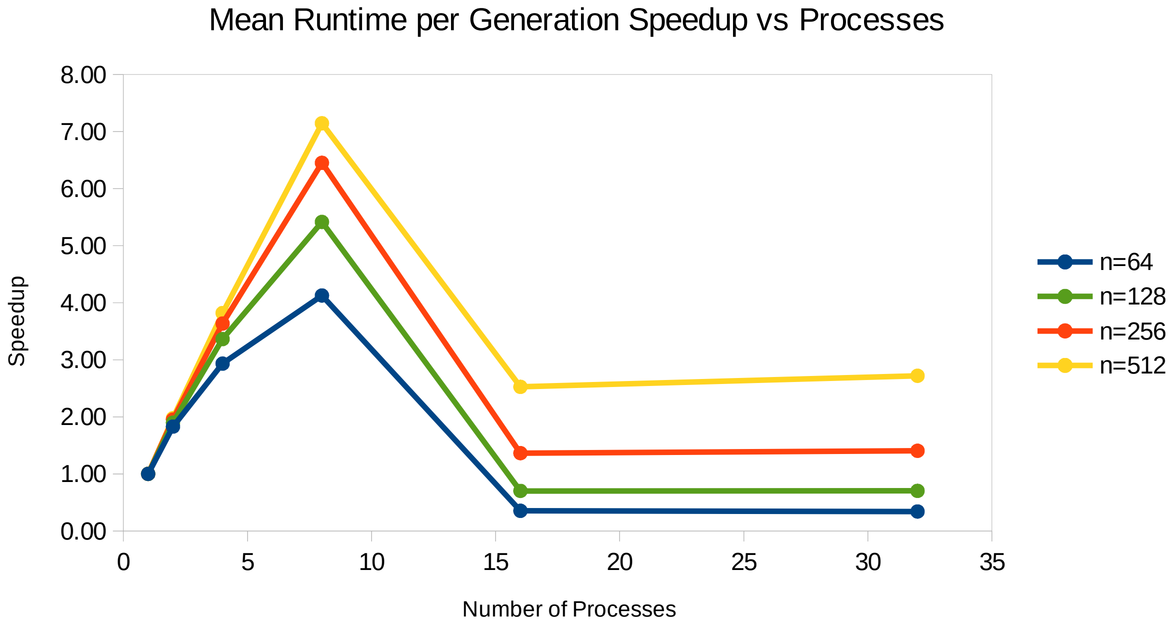


Table 4. Efficiency (S/p)

p	n=64	n=128	n=256	n=512
1	100.0	100.0	100.00	100.00
2	91.40	94.86	97.57	98.75
4	73.31	84.06	90.88	95.49
8	51.58	67.70	80.65	89.32
16	2.20	4.39	8.52	15.78
32	1.06	2.19	4.39	8.50

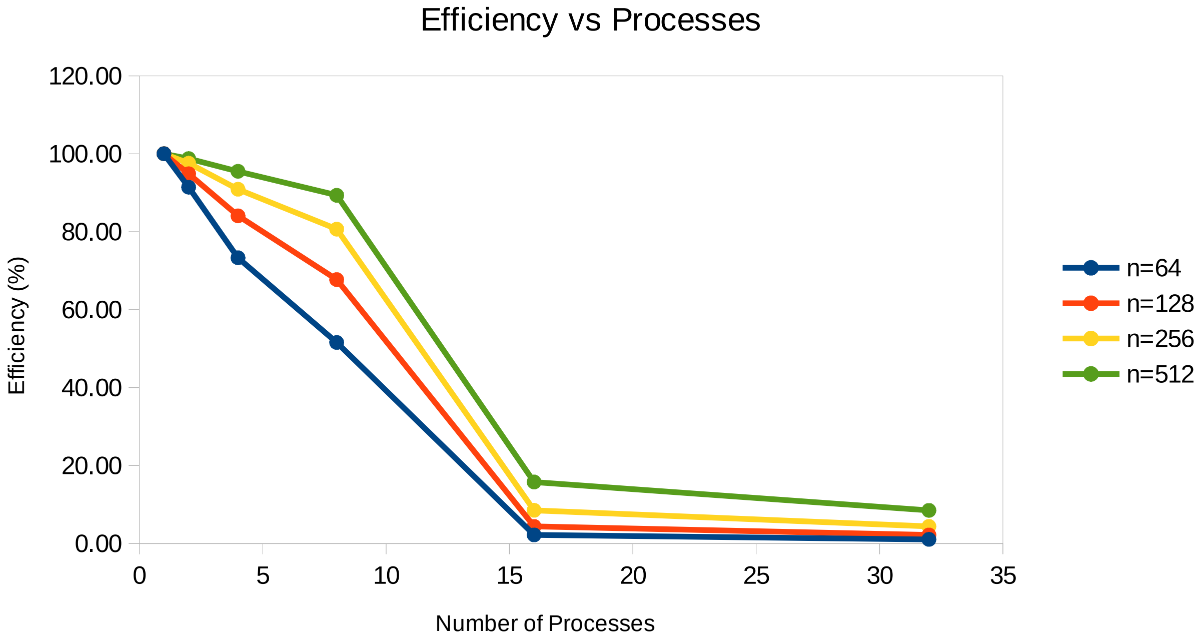
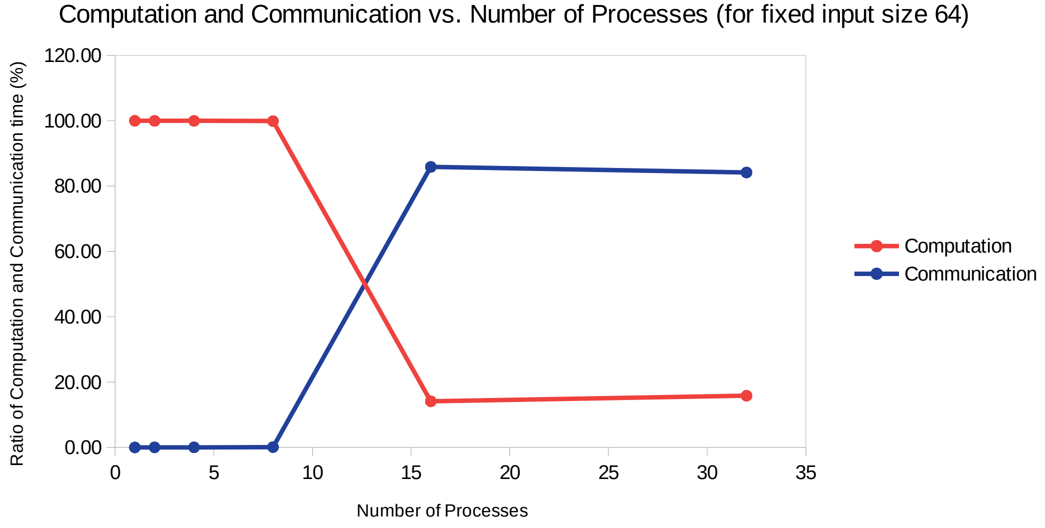


Table 5. Computation time / Total Run-time (%)

Gen = 10000	n=4	n=8	n=16	n=32	n=64	n=128	n=256	n=512
1	100.00	100.00	100.00	99.95	100.00	100.00	100.00	100.00
2	100.00	100.00	100.00	100.00	99.98	100.00	100.00	100.00
4	NA	100.00	100.00	99.89	99.96	99.97	99.99	100.00
8	NA	NA	100.00	99.87	99.90	99.93	99.96	99.98
16	NA	NA	NA	20.12	14.12	11.64	12.49	17.97
32	NA	NA	NA	NA	15.83	10.57	9.04	11.08
56	NA	NA	NA	NA	NA	NA	NA	NA

Table 6. Communication time / Total Run-time (%)

Gen = 10000	n=4	n=8	n=16	n=32	n=64	n=128	n=256	n=512
1	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00
4	NA	0.00	0.00	0.11	0.04	0.03	0.01	0.00
8	NA	NA	0.00	0.13	0.10	0.07	0.04	0.02
16	NA	NA	NA	79.88	85.88	88.36	87.51	82.03
32	NA	NA	NA	NA	84.17	89.43	90.96	88.92
56	NA	NA	NA	NA	NA	NA	NA	NA



Interpretation of Results:

I. Measure the average time per generation Chart:

Regarding the recorded average time per generation timing, for table 1. I chose a generation size of 10 however when I recorded these values my code was returning integers and we can observe that the smaller matrix sizes are very small and thus the program recorded it as zero milliseconds. Once I noticed that I needed to return a float to see the exact timing for each generation I changed my code to return floats. Table 2, in comparison to table 1, has a more precise values of the timing. But regardless of the table, both show are relatively similar pattern. Both tables showed an increase in time as the number of processes increased and input size increased. Additionally, once the number of nodes increased from 1 to 2, 4, etc. a drastic jump of timing happened. My assumption to this occurrence seems to be due to the latency of such processes. The more processes used the more time it will take to each processor to communicate with each other. Hence as the number of nodes (processes) are increased the average time for each generation will also increase.

II. Speedup:

The ideal relative speed up is a linear function, However, in the recorded Speedup pot for this experiment we can spot a sharp decrease in speedup across all input sizes. But before the sharp chance we can see that the speedup was ideal until 32 processes. Again I can conclude that as more processes needed to connect their own communication among each other, the larger the latency and therefore the longer the time it took for each generation to complete.

III. Efficiency:

Parallel efficiency (E) is a measure of how well the processors in the system are utilized. In practice, the efficiency curve tends to be a non-increasing (and more typically, decreasing) curve as more processors are added to the system in most real world applications. In this experiment we can spot a decreasing trend as the number of processes increase.

IV. Breakdown of Communication and Computation Run-times:

The graph demonstrates that the communication time percentage goes up as more processes are used. Which makes sense as more processes are taking more time to communicate with each other. The opposite happens to the Computation time percentage. As work is been divided by each processor and thus taking less time for the computation to happen.