Name – Joel Ebenezer Bellapurla

CSU ID – 2820745

Idea –

The code is written in python.

Modules used :

* Numpy – to generate the matrix and do the calculations
* multiprocessing (Method 2) – to do the parallel computing
* time – to get the running time

In method 1 and 2,

1. We read the power value from the user. Then the n is calculated by 2 ^ n (2\*\*n).
2. entries = np.random.randint(255, size=(n\*n)) – it generates random integers from 0 to 255 with a size of n\*n.
3. matrix\_x = np.array(entries).reshape(rows\_x, cols\_x) – this step creates the matrix with the entries from the previous step.
4. x\_transpose = matrix\_x.transpose() – this is used to get the transpose of the matrix.
5. The steps are repeated to create the Matrix Y
6. res = matrix\_x.dot(matrix\_x) + matrix\_x.dot(x\_transpose) + (matrix\_y.dot(matrix\_y)) \* 4 – this is used to compute X\*X + X\*XT + 4\*Y\*Y

In method 2,

Used multiprocessing module to parallelize the program.

1. Pool(processes=4) – this is initialized with n number of processors and we pass the fuction.
2. results = [pool.apply(res, args=(matrix\_x, x\_transpose, matrix\_y))] – used to run the function in parallel.

* res function computes X\*X + X\*XT + 4\*Y\*Y

1. pool.close() – is called when is the parallel function finishes execution.

Running time of Method 1 and Method 2

|  |  |  |
| --- | --- | --- |
| N values | Method 1 | Method 2 |
| n = 23 | 0.00022602081298828125 | 0.04540395736694336 |
| n = 24 | 0.00020503997802734375 | 0.034876108169555664 |
| n = 25 | 0.00037288665771484375 | 0.03410005569458008 |
| n = 26 | 0.001474142074584961 | 0.03298306465148926 |
| n = 27 | 0.009015798568725586 | 0.03770589828491211 |
| n = 28 | 0.060790061950683594 | 0.07932090759277344 |
| n = 29 | 0.3124051094055176 | 0.3301098346710205 |
| n = 210 | 3.6625308990478516 | 3.16184401512146 |
| n = 211 | 51.77370285987854 | 51.99432897567749 |
| n = 212 | 637.077931261597 | 795.6079001426697 |
| n = 213 | Taking too much time | Taking too much time |
| n = 214 | Taking too much time | Taking too much time |
| n = 215 | Taking too much time | Taking too much time |

Figure –

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated