INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DESIGN AND MANUFACTURING KANCHEEPURAM

LAB ASSIGNMENT 4 - REPORT
ON
VECTOR ADDITION
AND
VECTOR MULTIPLICATION
IN MPI

SUBMITTED BY

AMAR KUMAR

(CED17I029)

TO

DR. NOOR MAHAMMAD

VECTOR ADDITION

Strategy

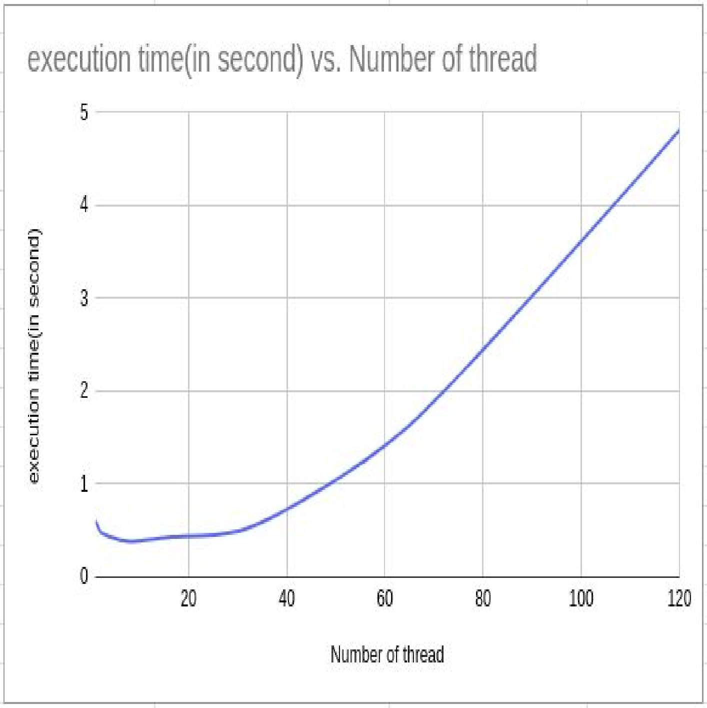
In my program for vector addition, the instruction which is running in parallel is in the "for" loop i.e **cp[i] = ap[i]+bp[i];**
Instead of running the program serially, we can distribute the task of adding vectors between master and worker so that my program could run parallely and in turn save the execution time.
Therefore , I have scattered a and b into ap and bp respectively. Also I have broadcasted the number of processes done by individual master or worker so that they only do vector addition of its own part only. After vector addition is done and stored in cp, I am gathering values of cp in c i.e. storing all individual cp to c.

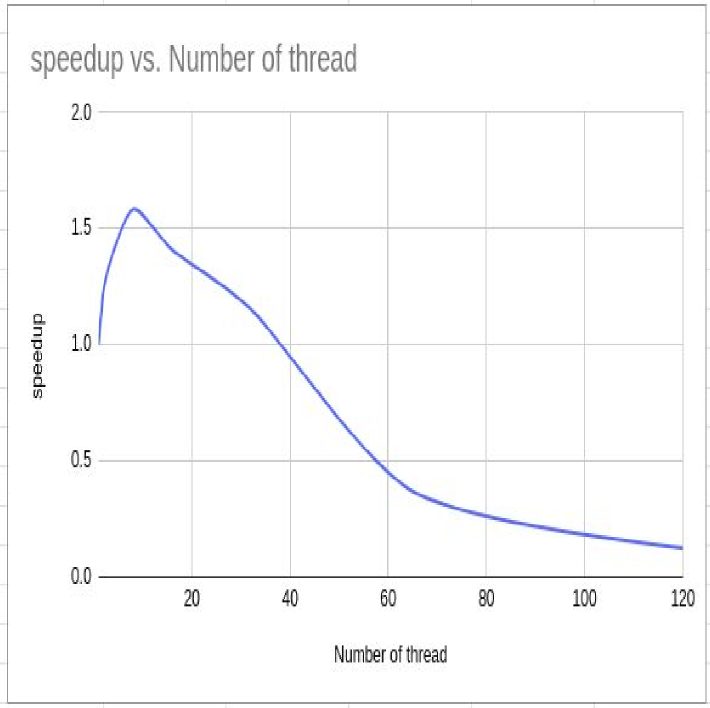
Graph and tables

https://docs.google.com/spreadsheets/d/1-wdrr1IYDmO5ER6nbt6rJzJLFmg2e0paNPbIuw6FzxY/edit?usp=sharing

Question2

Number of thread	execution time(in second)	speedup	parallelization fraction(f)
1	0.604755	1	0
2	0.490074	1.234007517	0.3792643302
4	0.431989	1.39993148	0.3809057662
8	0.381711	1.584326886	0.4215052849
16	0.429204	1.409015293	0.3096367951
32	0.523553	1.155097956	0.1386039295
64	1.585267	0.3814846332	-1.647073083
120	4.810509	0.1257153869	-7.012916752





Calculation of parallelization fraction

T(1) = 0.604755 seconds
Here , for P = 8 the execution time is minimum
T(P) = 0.381711 seconds

Speedup = $\frac{T(1)}{T(P)}$ = $\frac{0.604755}{0.381711}$ = 1.584326886

From Amdahl's Law,

Speedup = $\frac{1}{(f/P) + (1-f)}$ Where , f = Parallelization factor P = Thread Number

So, f = $\frac{(1-T(P)/T(1))}{(1-(1/P))}$

Therefore, f = 0.4215052849 which means that approx. 42% of the program is parallelizable.

VECTOR MULTIPLICATION

Strategy

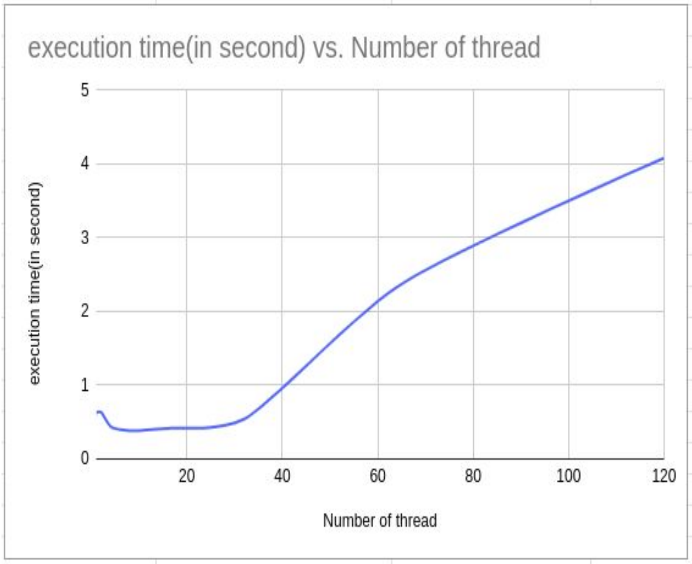
n my program for vector addition, the instruction which is running in parallel is in the "for" loop i.e **cp[i] = ap[i]*bp[i];**
Instead of running the program serially, we can distribute the task of multiplying vectors between master and worker so that my program could run parallely and in turn save the execution time.
Therefore , I have scattered a and b into ap and bp respectively. Also I have broadcasted the number of processes done by individual master or worker so that they only do vector multiplication of its own part only. After vector multiplication is done and stored in cp, I am gathering values of cp in c i.e. storing all individual cp to c

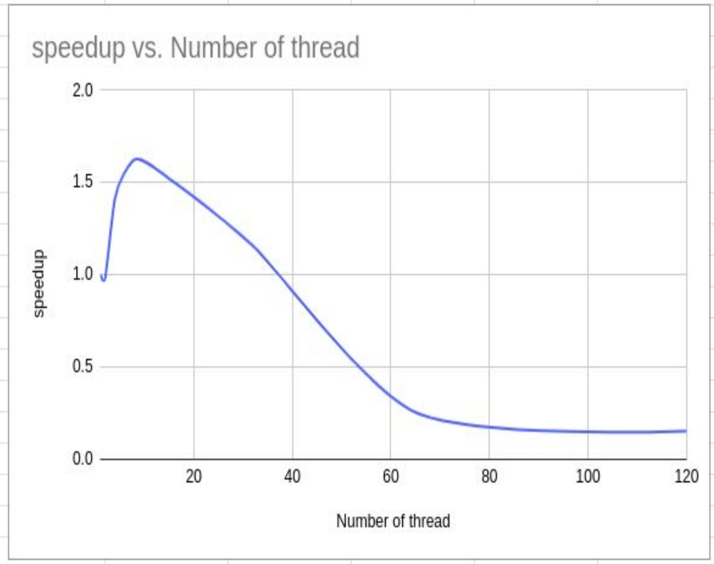
Graph and tables

https://docs.google.com/spreadsheets/d/1-wdrr1IYDmO5ER6nbt6rJzJLFmg2e0paNPbIuw6FzxY/edit?usp=sharing

Question3

Number of thread	execution time(in second)	speedup	parallelization fraction(f)
1	0.622951	1	0
2	0.635019	0.9809958442	-0.03874462036
4	0.440532	1.414087966	0.3904405536
8	0.383513	1.62432825	0.4392695871
16	0.414777	1.501893789	0.3564522196
32	0.538579	1.156656684	0.1398082312
64	2.331938	0.267138749	-2.786918679
120	4.080472	0.1526664072	-5.596870056





Calculation of parallelization fraction

T(1) = 0.622951 seconds
Here , for P = 8 the execution time is minimum
T(P) = 0.383513 seconds

Speedup = $\frac{T(1)}{T(P)}$ = $\frac{0.622951}{0.383513}$ = 1.62432825

From Amdahl's Law,

Speedup = $\frac{1}{(f/P) + (1-f)}$ Where , f = Parallelization factor P = Thread Number

So, f = $\frac{(1-T(P)/T(1))}{(1-(1/P))}$

Therefore, f = 0.4392695871 which means that approx. 44% of the program is parallelizable.