INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DESIGN AND MANUFACTURING KANCHEEPURAM

LAB ASSIGNMENT 6 - REPORT
ON
ADDITION OF N NUMBERS USING REDUCTION
AND VECTOR DOT PRODUCT

SUBMITTED BY
AMAR KUMAR
(CED17I029)
TO
DR. NOOR MAHAMMAD

ADDITION OF N NUMBERS

(USING REDUCTION)

Strategy

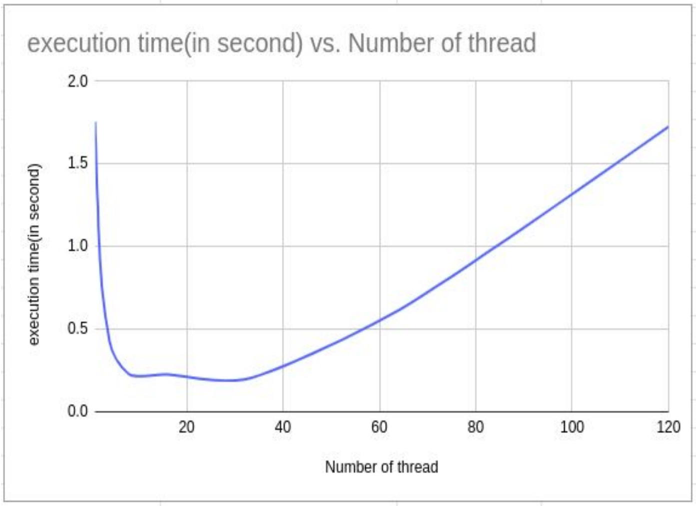
In my program for addition of n numbers, the instruction which is running in parallel is **psum +=numbers[i];**
Here i am using a reduction method to add these n numbers otherwise it should lead to racing conditions and thus should give wrong results.
What i did here is that i have distributed the work between master and workers by calculating the number of operations done by master and number of operations done by individual workers. All the distributed operations were done by each worker and master and stored in psum. Finally, MPI_Reduce was used to sum up all the individual elements of psum in sum using the function MPI_SUM inside MPI_Reduce.

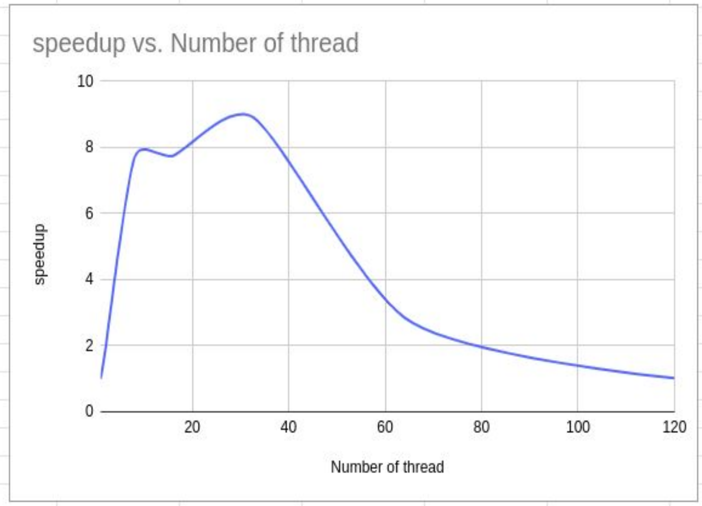
Graph and tables

https://docs.google.com/spreadsheets/d/1WaD36QLpFKjSw5OTOcGANxBBkhUXrizREsCHz8qaFeY/edit?usp=sharing

Question1

Number of thread	execution time(in second)	speedup	parallelization fraction(f)
1	1.754817	1	0
2	0.934036	1.878746644	0.9354605067
4	0.426447	4.114970911	1.009313222
8	0.228403	7.682985775	0.9941054497
16	0.226483	7.74811796	0.9289988263
32	0.195776	8.963391836	0.9170942868
64	0.615187	2.852493632	0.6597379471
120	1.725057	1.017251604	0.01710154622





Calculation of parallelization fraction

T(1) = 1.754817 seconds
Here , for P = 32 the execution time is minimum
T(P) = 0.195776 seconds

Speedup = $\frac{T(1)}{T(P)}$ = $\frac{1.754817}{0.195776}$ = 8.963391836

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 = 0.9170942868 which means that approx. 91% of the program is parallelizable.

VECTOR DOT PRODUCT

Strategy

In my program for vector dot product of n numbers, the instruction which is running in parallel is
dotproduct += a[i]*b[i];

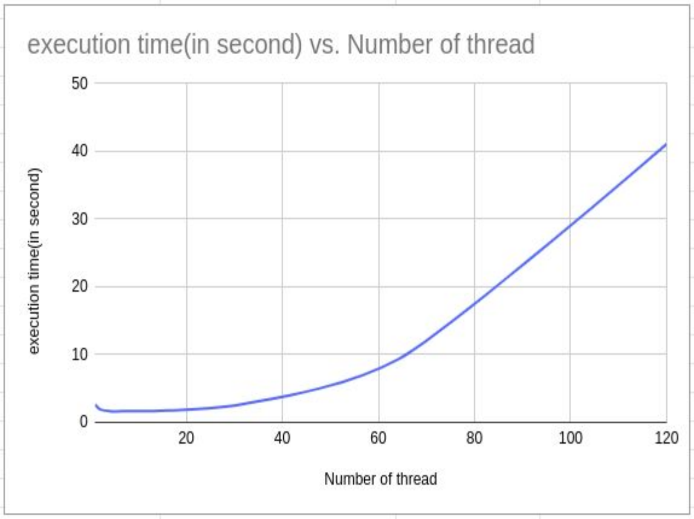
Here i am using a reduction method to multiply these two vectors.
What i did here is that i have distributed the work between master and workers by calculating the number of operations done by master and number of operations done by individual workers. All the distributed operations were done by each worker and master and stored in dotproduct. Finally, MPI_Reduce was used to sum up all the individual elements of dotproduct in result using the function MPI_SUM inside MPI_Reduce.

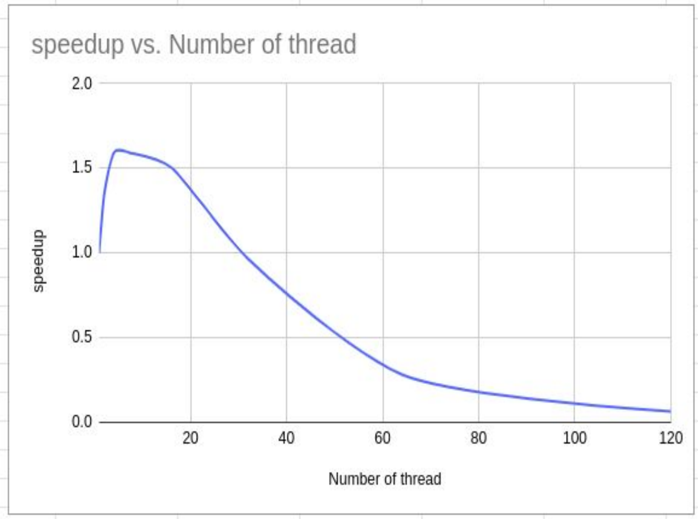
Graph and tables

https://docs.google.com/spreadsheets/d/1WaD36QLpFKjSw5OTOcGANxBBkhUXrizREsCHz8qaFeY/edit?usp=sharing

Question2

Number of thread	execution time(in second)	speedup	parallelization fraction(f)
1	2.611765	1	0
2	1.953578	1.336913602	0.5040170153
4	1.644123	1.588545991	0.4939913558
8	1.646142	1.586597633	0.4225376873
16	1.736241	1.504264097	0.3575713231
32	2.706862	0.9648681758	-0.0375855581
64	9.248994	0.2823836841	-2.581618883
120	41.061412	0.06360631242	-14.84542188





Calculation of parallelization fraction

T(1) = 2.611765 seconds
Here , for P = 4 the execution time is minimum
T(P) = 1.644123 seconds
Speedup = $\frac{T(1)}{T(P)}$ = $\frac{2.611765}{1.644123}$ = 1.588545991
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 = 0.4939913558 which means that approx. 49% of the program is parallelizable.