INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DESIGN AND MANUFACTURING KANCHEEPURAM

LAB ASSIGNMENT 7 - REPORT
ON
VECTOR ADDITION
AND
VECTOR MULTIPLICATION
IN CUDA

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 **c[i] = a[i] + b[i];**

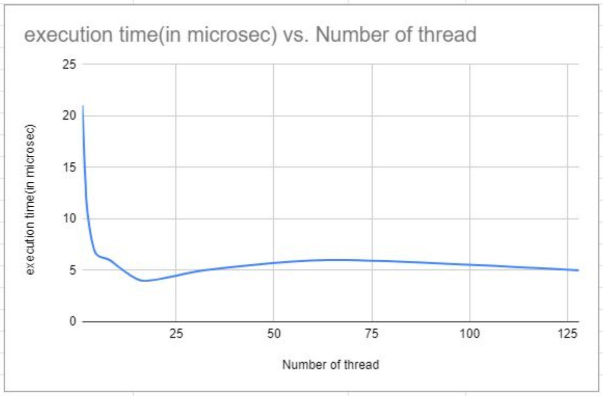
Instead of running the program serially, we can distribute the task of adding vectors between host and device so that my program could run parallely and in turn save the execution time.
Therefore, i have splitted the task of adding among threads i.e. if my vector size is 4 and number of threads are 2 then, thread1 will add 1st and 3rd element of vector and thread2 will add 2nd and 4th element of vector.

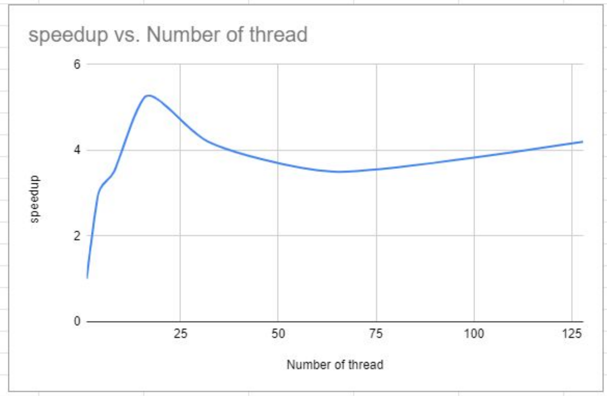
Graph and tables

https://docs.google.com/spreadsheets/d/11K-HmT02FA-CbqR2ENIR4bJBy9M8nUF5Jhf5b95zrJw/edit#gid=0

Question1

Number of thread	execution time(in microsec)	speedup	parallelization fraction(f)
1	21	1	0
2	12	1.75	0.8571428571
4	7	3	0.8888888889
8	6	3.5	0.8163265306
16	4	5.25	0.8634920635
32	5	4.2	0.7864823349
64	6	3.5	0.7256235828
128	5	4.2	0.767904012
256	6	3.5	0.7170868347
500	5	4.2	0.7634316252





Calculation of parallelization fraction

T(1) = 21 microseconds
Here , for P = 16 the execution time is minimum
T(P) = 4 microseconds
Speedup = $\frac{T(1)}{T(P)}$ = $\frac{21}{4}$ = 5.25
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 = 0.8634920635 which means that approx. 86% of the program is parallelizable.

<u>VECTOR MULTIPLICATION</u>

Strategy

In my program for vector addition, the instruction which is running in parallel is in the "for" loop i.e **c[i] = a[i] * b[i];**

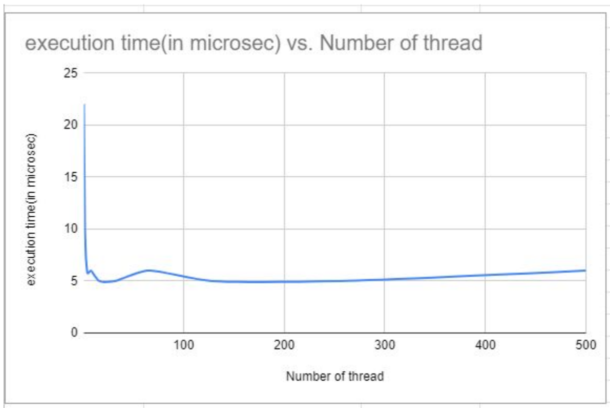
Instead of running the program serially, we can distribute the task of adding multiplying between host and device so that my program could run parallely and in turn save the execution time.
Therefore, i have splitted the task of multiplying among threads i.e. if my vector size is 4 and number of threads are 2 then, thread1 will multiply 1st and 3rd element of vector and thread2 will multiply 2nd and 4th element of vector.

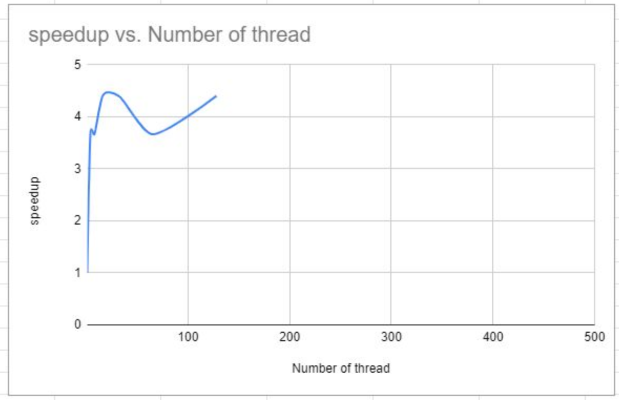
Graph and tables

https://docs.google.com/spreadsheets/d/11K-HmT02FA-CbqR2ENIR4bJBy9M8nUF5Jhf5b95zrJw/edit#gid=0

Question2

Number of thread	execution time(in microsec)	speedup	parallelization fraction(f)
1	22	1	0
2	10	2.2	1.090909091
4	6	3.666666667	0.9696969697
8	6	3.666666667	0.8311688312
16	5	4.4	0.8242424242
32	5	4.4	0.7976539589
64	6	3.666666667	0.7388167388
128	5	4.4	0.7788117394
256	5	4.4	0.7575757575758
500	6	3.666666667	0.7287301876





Calculation of parallelization fraction

T(1) = 22 microseconds
Here , for P = 16 the execution time is minimum
T(P) = 5 microseconds
Speedup = $\frac{T(1)}{T(P)}$ = $\frac{22}{5}$ = 4.4
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 = 0.8242424242 which means that approx. 82% of the program is parallelizable.