**PARALLEL AND DISTRIBUTED COMPUTING**

**LAB 1**

S Shyam Sundaram

19BCE1560

14th August, 2021

**QUESTION**

Compare the time taken when vector addition is serialized and parallelized.

**CODE**

#include <stdio.h>

#include "omp.h"

#include<time.h>

#define N 10000

int main()

{

float a[N],b[N],c[N];

int i;

float start,end,exec;

for(i=0;i<N;++i)

{

a[i]=(i+1)\*1.0;

b[i]=(i+1)\*2.0;

}

start=omp\_get\_wtime();

#pragma omp parallel default(none), private(i), shared(a,b,c)

{

#pragma omp for

for(i=0;i<N;++i)

{

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

}

}

end=omp\_get\_wtime();

exec=end-start;

printf("Time taken is: %f\n",exec);

return 0;

}

**EXECUTION**

commands:

export OMP\_NUM\_THREADS=2

gcc -fopenmp prog.c

./a.out

*Note*: For Sequential, OMP\_NUM\_THREADS=1

**OUTPUT**

Time taken by sequential execution and parallel execution with varying number of threads are recorded below.

|  |  |  |  |
| --- | --- | --- | --- |
| Size of the Vector | Time for Sequential Execution | Number of threads | Time for Parallel Execution |
| 1000 | 0.000002 | 2 | 0.000244 |
| 1000 | 4 | 0.000244 |
| 1000 | 8 | 0.000244 |
| 10000 | 0.000000 | 2 | 0.000244 |
| 10000 | 4 | 0.000244 |
| 10000 | 8 | 0.000488 |
| 100000 | 0.000629 | 2 | 0.000506 |
| 100000 | 4 | 0.000458 |
| 100000 | 8 | 0.003638 |
| 600000 | 0.007080 | 2 | 0.003662 |
| 600000 | 4 | 0.002441 |
| 600000 | 8 | 0.005616 |