High Performance Computing

COM403P

Experiment-1

Vector Addition

Kadali Paul Babu

CED19I002

CED19I002
Kadali Paul Babu

Objective

Parallelize the Vector Addition for given N double precision floating point numbers.

Serial Code

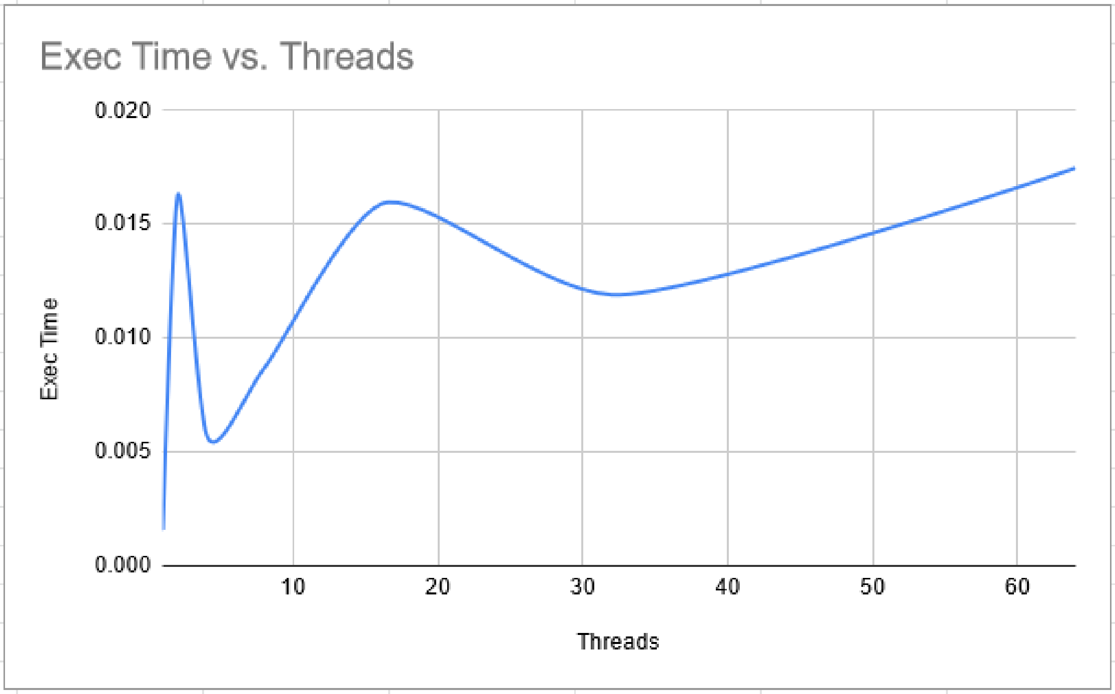
```
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <time.h>

#define SIZE 100000

int main()
{
    double a[SIZE], b[SIZE], c[SIZE], rand_a, rand_b;
    double start, end, exec;

    start = omp_get_wtime();

    for (int i = 0; i < SIZE; i++)
    {
        rand_a = rand();
        rand_b = rand();

        a[i] = i*rand_a;
        b[i] = i*rand_b;

        for(int j = 1; j < SIZE; j++)
        c[i] = a[i] + b[i];
    }

    end = omp_get_wtime();

    exec = end - start;

    printf("Serial Exec time - %f\n", exec);

    return 0;
}
```

Parallel Code

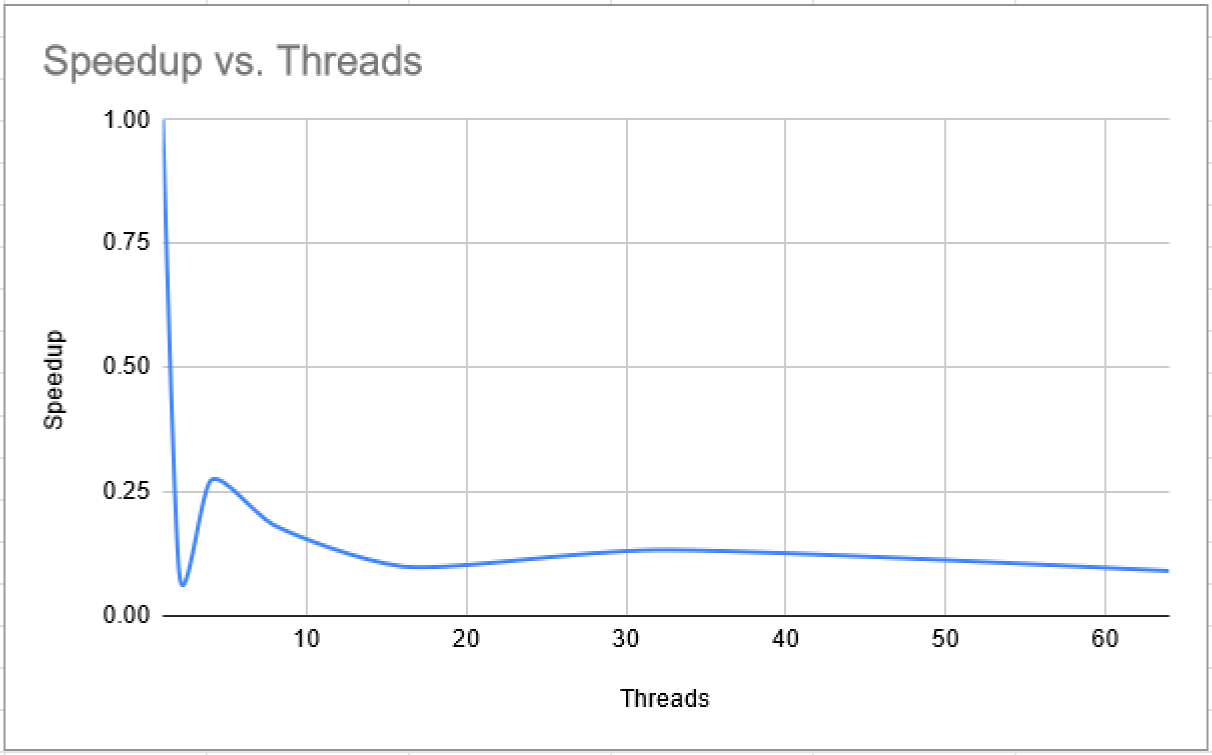
```
#include <stdio.h>
#include <stdlib.h>
```

```
#include <omp.h>
#include <time.h>

#define SIZE 100000

int main()
{
    double a[SIZE], b[SIZE], c[SIZE], rand_a, rand_b;
    double start, end, exec;

    start = omp_get_wtime();

    for (int i = 0; i < SIZE; i++)
    {
        rand_a = rand();
        rand_b = rand();

        a[i] = i*rand_a;
        b[i] = i*rand_b;

        for(int j = 1; j < SIZE; j++)
        c[i] = a[i] + b[i];
    }

    end = omp_get_wtime();

    exec = end - start;

    printf("Serial Exec time - %f\n", exec);

    return 0;
}
```

Threads vs Time

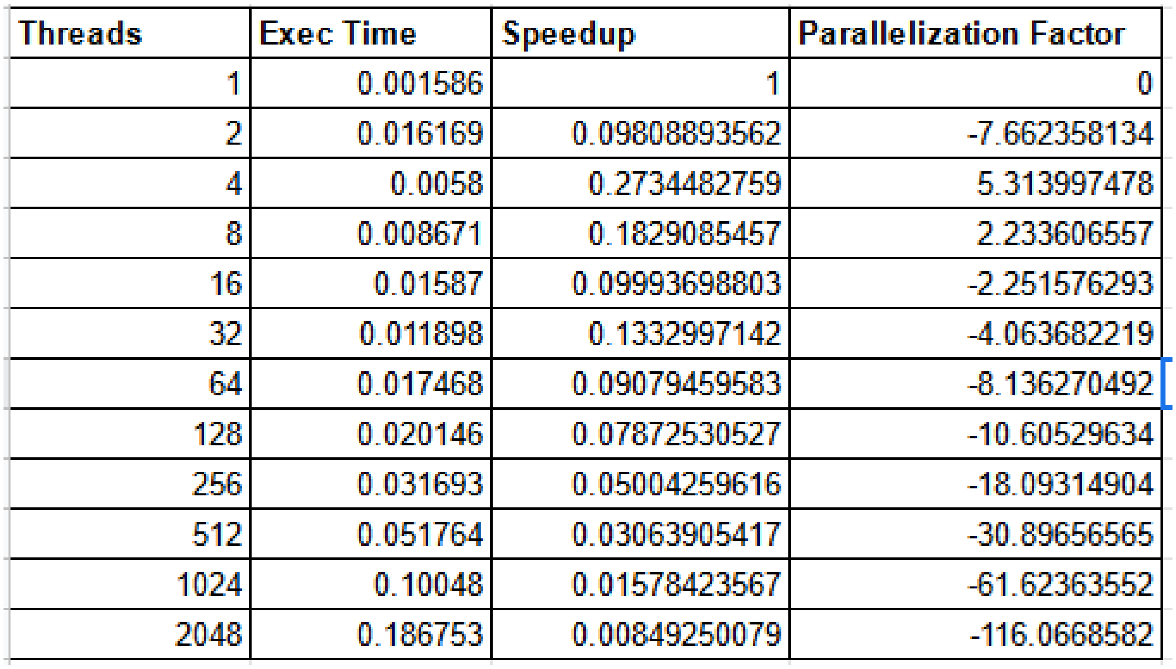


Speedups vs Threads



Parallelization factor

Formula for calculating the Parallelization Factor = (1-(1/speedup))*(1-(1/p))
where, p = number of threads/processor



Threads	Exec Time	Speedup	Parallelization Factor
1	0.001586	1	0
2	0.016169	0.09808893562	-7.662358134
4	0.0058	0.2734482759	5.313997478
8	0.008671	0.1829085457	2.233606557
16	0.01587	0.09993698803	-2.251576293
32	0.011898	0.1332997142	-4.063682219
64	0.017468	0.09079459583	-8.136270492
128	0.020146	0.07872530527	-10.60529634
256	0.031693	0.05004259616	-18.09314904
512	0.051764	0.03063905417	-30.89656565
1024	0.10048	0.01578423567	-61.62363552
2048	0.186753	0.00849250079	-116.0668582

Inferences

From the above graphs and data we can clearly observe that the execution time decreases until a particular point and then increases as we add more threads to run the process.Using many threads leads to more context switches which lowers the speed up. We can conclude that the maximum parallelization happens at 4 threads