Sreepathy Jayanand

CED17I038

High Performance Computing Lab - 7

Analysis of parallelization factor

No of Grids used = 1

No of blocks used per grid = 1

No of threads per block = 1, 2, 4, 6, 8, 10, 12, 14, 16, 20, 24

Problem statement 1 : Given 2 arrays A & B find C such that find C such that C[i] = A[i] + B[i]

The arrays have size ‘n’ = 1e8

Let the number of threads doing the job = t,

Strategy: Each thread gets ceil( n / t) elements to compute the sum of A and B.

For thread 0 : Indices which it will calculate the sum of is : 0, t, 2t, 3t …

i.e A[0] + B[0] = C[0], A[t] + B[t] = C[t], A[2t] + B[2t] = C[2t] ….

For thread 1 : Indices which it will calculate the sum of is : 1, t + 1, 2t + 1. 3t + 1….

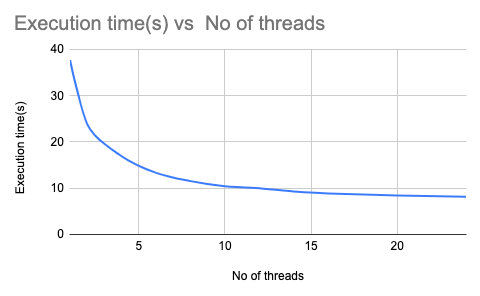
i.e A[1] + B[1] = C[1], A[t + 1] + B[t + 1] = C[t + 1], A[2t + 1] + B[2t + 1] = C[2t + 1] ….

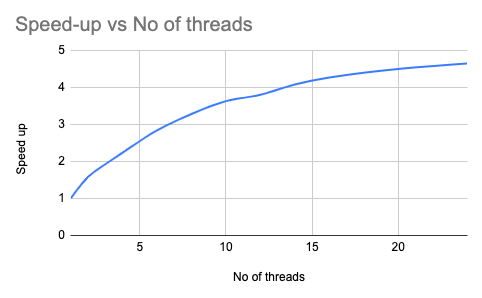
And so on…

Table for vector addition:

| Number of Threads | System Time taken for execution(s) |
| --- | --- |
| 1 | 37.8 |
| 2 | 23.9 |
| 4 | 16.9 |
| 6 | 13.3 |
| 8 | 11.5 |
| 10 | 10.4 |
| 12 | 9.93 |
| 14 | 9.25 |
| 16 | 8.84 |
| 20 | 8.39 |
| 24 | 8.12 |

Plots:





The most efficient thread size for this program is 24

According to Amdahl’s law :

Speed up = 1/((1-p) + p/N)

Or

Parallelization factor = (N - N/S.U)/(N - 1)

So here N = 24;

Speed Up = 37.8/8.12 = 4.65

PF = (24 - 24/4.65)/23 = 0.81

Problem statement 2 : Given 2 arrays A & B find C such that C[i] = A[i] \* B[i]

No of Grids used = 1

No of blocks used per grid = 1

No of threads per block = 1, 2, 4, 6, 8, 10, 12, 14, 16, 20, 24

The arrays have size ‘n’ = 1e8

Let the number of threads doing the job = t,

Strategy: Each thread gets ceil( n / t) elements to compute the sum of A and B.

For thread 0 : Indices which it will calculate the sum of is : 0, t, 2t, 3t …

i.e A[0] \* B[0] = C[0], A[t] \* B[t] = C[t], A[2t] \* B[2t] = C[2t] ….

For thread 1 : Indices which it will calculate the sum of is : 1, t + 1, 2t + 1. 3t + 1….

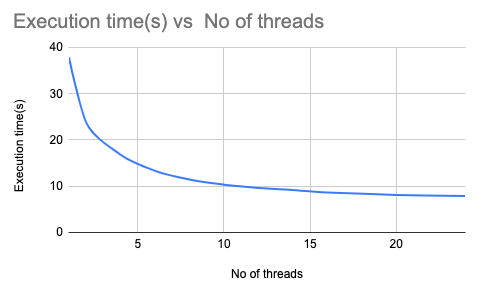
i.e A[1] \* B[1] = C[1], A[t + 1] \* B[t + 1] = C[t + 1], A[2t + 1] \* B[2t + 1] = C[2t + 1] ….

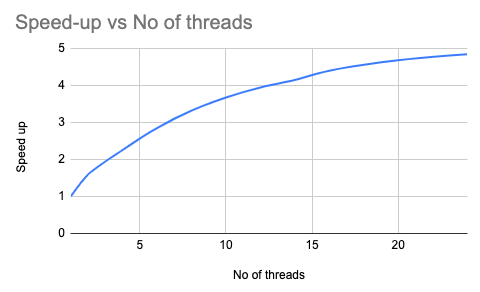
And so on…

Table for vector multiplication:

| Number of Threads | System Time taken for execution(s) |
| --- | --- |
| 1 | 37.9 |
| 2 | 23.8 |
| 4 | 16.8 |
| 6 | 13.3 |
| 8 | 11.4 |
| 10 | 10.3 |
| 12 | 9.59 |
| 14 | 9.12 |
| 16 | 8.6 |
| 20 | 8.08 |
| 24 | 7.81 |

Plots:





The most efficient thread size for this program is 24

According to Amdahl’s law :

Speed up = 1/((1-p) + p/N)

Or

Parallelization factor = (N - N/S.U)/(N - 1)

So here N = 24;

Speed Up = 37.9/7.81 = 4.85

PF = (24 - 24/4.85)/23 = 0.82