

Practical No. 3

Study and Implementation of schedule, nowait, reduction, ordered and collapse clauses

Q1: Analyse and implement a Parallel code for below program using OpenMP.

// C Program to find the minimum scalar product of two vectors (dot product)

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

```
int sort(int arr[], int n)
```

```
{
```

```
    int i, j;
```

```
    for (i = 0; i < n-1; i++)
```

```
        for (j = 0; j < n-i-1; j++)
```

```
            if (arr[j] > arr[j+1])
```

```
                {
```

```
                    int temp = arr[j];
```

```
                    arr[j] = arr[j+1];
```

```
                    arr[j+1] = temp;
```

```
                }
```

```
    }
```

```
int sort_des(int arr[], int n)
```

```
{
```

```
    int i,j;
```

```
    for (i = 0; i < n; ++i)
    {
        for (j = i + 1; j < n; ++j)
        {
            if (arr[i] < arr[j])
            {
                int a = arr[i];
                arr[i] = arr[j];
                arr[j] = a;
            }
        }
    }
}
```

```
int main()
{
    //fill the code;

    int n;

    scanf("%d",&n);

    int arr1[n], arr2[n];

    int i;

    for(i = 0; i < n ; i++)
```

```
        {  
            scanf("%d",&arr1[i]);  
        }  
    for(i = 0; i < n ; i++)  
    {  
        scanf("%d",&arr2[i]);  
    }  
  
    sort(arr1, n);  
    sort_des(arr2, n);  
    int sum = 0;  
    for(i = 0; i < n ; i++)  
    {  
        sum = sum + (arr1[i] * arr2[i]);  
    }  
    printf("%d",sum);  
    return 0;  
}
```

Q2: Write OpenMP code for two 2D Matrix addition, vary the size of your matrices from 250, 500, 750, 1000, and 2000 and measure the runtime with one thread (Use functions in C in calculate the execution time or use GPROF)

- i. For each matrix size, change the number of threads from 2,4,8., and plot the speedup versus the number of threads.**
- ii. Explain whether or not the scaling behaviour is as expected.**

Q3. For 1D Vector (size=200) and scalar addition, Write a OpenMP code with the following:

- i. Use STATIC schedule and set the loop iteration chunk size to various sizes when changing the size of your matrix. Analyze the speedup.**
- ii. Use DYNAMIC schedule and set the loop iteration chunk size to various sizes when changing the size of your matrix. Analyze the speedup.**
- iii. Demonstrate the use of nowait clause.**