

PARALLEL AND DISTRIBUTED COMPUTING – CSE4001

LAB ASSIGNMENT- 1

Name: Sagar S

Reg No: 19BCE2460

Lab slot: 45+46

Faculty: Prof. Balamurugan R

Q1) Using OpenMP, Design, develop and run a multi-threaded program to perform and print vector addition.

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

#define ARRAY_SIZE 8
#define NUM_THREADS 4

int main(int argc, char *argv[])
{

    int *a;
    int *b;
    int *c;

    int n = ARRAY_SIZE;
    int n_per_thread;
    int total_threads = NUM_THREADS;
    int i;
    a = (int *)malloc(sizeof(int) * n);
    b = (int *)malloc(sizeof(int) * n);
    c = (int *)malloc(sizeof(int) * n);
```

```

    for (i = 0; i < n; i++)
    {
        a[i] = i;
    }
    for (i = 0; i < n; i++)
    {
        b[i] = i;
    }

    omp_set_num_threads(total_threads);

    n_per_thread = n / total_threads;

#pragma omp parallel for shared(a, b, c) private(i) schedule(static, n_per_thread)
    for (i = 0; i < n; i++)
    {
        c[i] = a[i] + b[i];
        printf("Thread %d works on element%d\n", omp_get_thread_num(), i);
    }

    printf("i\ta[i]\t+\tb[i]\t=\tc[i]\n");
    for (i = 0; i < n; i++)
    {
        printf("%d\t%d\t+\t%d\t=\t%d\n", i, a[i], b[i], c[i]);
    }

    free(a);
    free(b);
    free(c);

    return 0;
}

```

```
ss007@ss007-VirtualBox: ~/PDC DAs
ss007@ss007-VirtualBox:~/PDC DAs$ export OMP_NUM_THREADS=4
ss007@ss007-VirtualBox:~/PDC DAs$ gcc -o da1-1a -fopenmp da1-1a.c
ss007@ss007-VirtualBox:~/PDC DAs$ ./da1-1a
Thread 1 works on element2
Thread 1 works on element3
Thread 0 works on element0
Thread 0 works on element1
Thread 2 works on element4
Thread 2 works on element5
Thread 3 works on element6
Thread 3 works on element7
i      a[i]  +      b[i]  =      c[i]
0      0      0      0
1      1      1      2
2      2      2      4
3      3      3      6
4      4      4      8
5      5      5     10
6      6      6     12
7      7      7     14
ss007@ss007-VirtualBox:~/PDC DAs$
```

Q2) Using OpenMP, Design, develop and run a multi-threaded program to perform sum of N elements (N=2021) using Loop work Sharing with schedule clause.

```
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
#define N 2021
int main (int argc, char *argv[])
{

int i, nthreads, tid;
int a[N], sum = 0 ;

for (i=0; i<=N; i++)
{
a[i] = i ;
}

#pragma omp parallel shared(a,nthreads) private(i,tid)
```

```

{
tid = omp_get_thread_num();
if (tid == 0)
{
nthreads = omp_get_num_threads();
printf("Number of threads = %d\n", nthreads);
}
printf("Thread %d starting...\n",tid);
#pragma omp for
for (i=0; i<=N; i++)
{
sum = sum + a[i];
if((i%100)==0)
printf("Thread %d, Current Sum Value = %d\n",tid,sum);
if(i==2021)
printf("Thread %d updated Last Value which is %d\n",tid,sum);
}
}

printf("\nSum of elements of Vector 0 with 2021 elements is %d\n ", sum);
}

```

```

ss007@ss007-VirtualBox: ~/PDC DAs
ss007@ss007-VirtualBox:~/PDC DAs$ gcc -o dai-1b -fopenmp dai-1b.c
ss007@ss007-VirtualBox:~/PDC DAs$ ./dai-1b
Number of threads = 4
Thread 0 starting...
Thread 0, Current Sum Value = 0
Thread 0, Current Sum Value = 5050
Thread 0, Current Sum Value = 20100
Thread 0, Current Sum Value = 45150
Thread 0, Current Sum Value = 80200
Thread 0, Current Sum Value = 125250
Thread 3 starting...
Thread 3, Current Sum Value = 258679
Thread 3, Current Sum Value = 423729
Thread 3, Current Sum Value = 598779
Thread 3, Current Sum Value = 783829
Thread 3, Current Sum Value = 978879
Thread 3 updated Last Value which is 1021110
Thread 2 starting...
Thread 2, Current Sum Value = 1115094
Thread 2, Current Sum Value = 1238144
Thread 2, Current Sum Value = 1355194
Thread 2, Current Sum Value = 1498244
Thread 2, Current Sum Value = 1635294
Thread 1 starting...
Thread 1, Current Sum Value = 1711965
Thread 1, Current Sum Value = 1777015
Thread 1, Current Sum Value = 1852065
Thread 1, Current Sum Value = 1937115
Thread 1, Current Sum Value = 2032165
Sum of elements of Vector 0 with 2021 elements is 2043231
ss007@ss007-VirtualBox:~/PDC DAs$

```