Name: Raghav Maheshwari
Roll No: 53
Panel: A

Lab Assignment - 2 (PP)

* Problem statement:
Write a program for large vector addition, Convert serial vector addition program to parallel vector addition. Measure time taken for serial vs parallel vector addition for various sizes of computation. Plot a graph of execution times (T_s and T_p) to identify threshold for which you start getting speed up.

* AIM:
To write a C program to add two large vectors and run program on a multi-core parallel system.

* OBJECTIVE:
To understand conversion of serial code to parallel work.

* THEORY:
Write Complexity analysis of vector addition and parallel vector addition.

Vector addition systems are an important model in theoritical computer science and have been used in a variety of areas. For these systems, we are interested in standard

notion of computational time complexity i.e. we want to understand length of longest tower for a fixed vector addition system with states depending on size of initial configuration. We show that asymptotic complexity of a given vector addition system with states is either $O(n^k)$ for some computable integer k, where n is size of initial configuration, or at least exponential. Finally we show that $1 \leq k \leq 2^n$ where 'n' is the dimension of considered vector addition system.

* CONCLUSION:
Successfully implemented serial to parallel conversion of vector addition code and analyze speedup and efficiency.

* FAQ

1) Name the top three current supercomputer of the world with their specification.

Ans : Fugaku.
CPU cores — 7,630,848 (158,976 × 48 core Fujitsu A64fx @ 2.2 GHz)
Accelerator cores — 0
Interconnect — Tofu interconnect D.
OS — Linux (RHEL)
Model — Supercomputer Fugaku

ii) Summit:

CPU cores - 202,752 (9,216 x 22- core IBM POWER9 @ 3.07GH

Accelerator cores - 27,648 x 80 Nvidia Tesla V100.

Interconnect - InfiniBand EDR

OS - Linux (RHEL 7.4)

Model - IBM Power System AC922

iii) Sierra

CPU cores - 190,080 (8,640 x 22 core IBM POWER 29 @ 3.1G

Accelerator cores - 172,80 x 80 Nvidia Tesla V100.

Interconnect - InfiniBand EDR

OS - Linux (RHEL)

Model - IBM Power System S922LC.

2) What is Moore's Test?

Ans Moore's ~~Test~~ Law states that number of transistors on a microchip doubles about every two years, the cost of computers is halved.

Code:

```
#include <stdio.h>
#include <omp.h>
#include <stdlib.h>
double parallel(int n){
int *a,*b,*c;
a = (int *)malloc(sizeof(int)*n);
b = (int *)malloc(sizeof(int)*n);
c = (int *)malloc(sizeof(int)*n);
for(int i=0; i<n;i++)
{
a[i] = rand()%n;
b[i] = rand()%n;
}
double t_startp = omp_get_wtime();
#pragma omp parallelfor private(i)
for(int i=0; i<n; i++)
{
c[i]=a[i]+b[i];
}
double t_endp = omp_get_wtime();
double timep = t_endp - t_startp;
return timep;
}
double serial(int n){
int *a,*b,*c;
```

```
a = (int *)malloc(sizeof(int)*n);
b = (int *)malloc(sizeof(int)*n);
c = (int *)malloc(sizeof(int)*n);
for(int i=0; i<n;i++)
{
a[i] = rand()%n;
}
for(int i=0; i<n;i++)
{
b[i] = rand()%n;
}
double t_starts = omp_get_wtime();
for(int i=0; i<n; i++)
{
c[i]=a[i]+b[i];
}
double t_ends = omp_get_wtime();
double times = t_ends - t_starts;
return times;
}
int main()
{
int t;
printf("Enter the number of tests:");
scanf("%d",&t);
int n[t];
```

```
for(int i=0;i<t;i++){
printf("Enter the value:");
scanf("%d",&n[i]);
}
printf("Data Point \t Time taken for Serial Approach \t Time taken for
Parallel Approach\n");
for(int i=0;i<t;i++){
printf("%d\t\t\t%lf\t\t\t\t%lf\n",n[i],serial(n[i]),parallel(n[i]));
}
}
```

Output: