PARTHIVI CHOUBEY R.no. - 60

180905456 CSE - B - VI

PAGE NO. 1

DATE 23/04/2021

PCAP Assignment - 1

Parthivi Choubey

Q2.

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

int main (int argc, char *argv[])
{
    int size, rank, ~~arr[]~~; mat[4][4];
    int *arr, int n, crr[4];
    MPI_init (&argc, &argv);
    MPI_Comm_rank (MPI_COMM_WORLD, &rank);
    MPI_Comm_size (MPI_COMM_WORLD, &size);
    if (rank == 0)
    {
        printf ("Enter size of 1D array: ");
        scanf ("%d", &n);
        arr = malloc (n * sizeof (int));
        printf ("Enter value for 1D array :\n");
        for (int i=0; i<n; i++)
            scanf ("%d", &arr[i]);
        printf ("Enter value for 2D matrix:\n");
        for (int i=0; i<4; i++)
            for (int j=0; j<4; j++)
                scanf ("%d", &mat[i][j]);
    }   n = n/4;
    ~~MPI_Scatter (arr,~~
    ~~recv~~ int brr[n];
    MPI_Scatter (arr, n, MPI_INT, brr, n,
                 MPI_INT, 0, MPI_COMM_WORLD);
    int min = brr[0];
    MPI_Bcast (&n, 1, MPI_INT, 0, MPI_COMM_
               WORLD);
```

```
for (int i=1; i<n; i++)
    if (min > brr[i])
        min = brr[i];
MPI_Scatter (mat, 4, MPI_INT, crr, 4,
    MPI_INT, 0, MPI_COMM_WORLD);
int man = crr[0];
for (int i = 1; i < 4; i++)
    if (man < crr[i])
        man = crr[i];
printf ("Rank = %d
        Minimum = %d ", rank, min);
printf ("Min in array + Max in
    column %d in rank %d = %d ",
    rank, rank, (min + man[rank]));
MPI_Finalize ();
return 0;
}
```

```
int man[4];
MPI_Reduce (crr, man, 1,
    MPI_INT, MPI_MIN, 0, MPI_COMM_
                        WORLD);
```

Q1. ~~To~~ ~~important~~ CPU design:
The design of a CPU is optimized for sequential code performance. It makes use of sophisticated control logic to allow ins from a single thread of execution to execute in ||. The large cache memories are provided to reduce the ins & data access latencies of large complex applications → latency oriented design. Memory bandwidth limit the speed of apps by limiting the rate at which data can be delivered from the memory system to processors.

GPU design:
The design is shaped by gaming industry, ability to perform a huge no. of ~~ops~~ floating pt. calculations / video frame. GPU performs well by executing massive no. of threads. GPU h/w takes adv. of a large no. of execution threads to find work to do when some of them are waiting for long latency memory accesses. Small cache memories are provided to help with bandwidth requirements of apps, so multiple threads that access same memory data need not always access DRAM → throughput oriented design