Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

Student Answer Script View

MIT MPL - 1st-3rd-5th and 7th Semester - Mid Term Examination - September 2023 **Answer Sheet**

Student Name: HARIKA BOPPANA . .

Roll Number: 210962030

Course: Computer Science and Engineering - Artificial Intelligence and Machine Learning

Year/Sem: Semester 5

Subject Name: PARALLEL COMPUTER ARCHITECTURE AND PROGRAMMING

Exam Date: 27-Sep-2023

21.00 / 30.00

Q.No : 1) **Score : 0.00 / 0.50**

Apply convolution filter [1, 2, 3, 2, 1] for element 1 for the array [1, 2, 3, 4, 5, 6,7]. Your answer is _____________.

10 | 20 | 27 | 9

Q.No : 2) **Score : 0.00 / 0.50**

Choose the full form of CUDA

Compute Unified Device Architecture | Collective Unified Device Architecture | Compute Unified Data Architecture | Collective Unified Data Architecture

Q.No : 3) **Score : 0.50 / 0.50**

ENG US 1:15 pm 14/10/2023

Q.No : 3) Score : 0.50 / 0.50

Compute the threadId of the thread belonging to block2 thread3 if the block dimension is 32 and the grid dimension is 16.
Your answer is ___________________

35 | 67 | 50 | 98

Q.No : 4) Score : 0.00 / 0.50

Choose the correct answer

A)The convolution filter size is larger than the data size.

B)The number of grids is equivalent to the number of kernels.

T F | F F | F T | T T

Q.No : 5) Score : 0.50 / 0.50

.Identify the correct statement with respect to MPI programming.

A)Collective communication routines must involve all processes within the scope of a communicator.

B)MPI_Barrier function will increase parallelism.

F F | F T | T F | T T

Q.No : 6) Score : 0.00 / 0.50

Compared to shared-variable programs, MPI programs tend to exhibit ____ *cache hit rates* when executing on

Q.No : 6) Score : 0.00 / 0.50

Compared to shared-variable programs, MPI programs tend to exhibit ____ *cache hit rates* when executing on multiprocessors and d*ebugging them is* _____ than debugging shared-variable programs.

high, simpler | low, simpler | high, difficult | low, difficult

Q.No : 7) Score : 0.50 / 0.50

Identify the correct statement with respect to parallel pipeline processing.

A)It is a combination of temporal and data parallelism.

B)Load balancing of work is possible in static assignment

F T | T F | F F | T T

Q.No : 8) Score : 0.00 / 0.50

Values of integer array A is as shown below in 3 processes in which process 0 being the root:

In Process 0: 12 34 56 32 15 20

In Process 1: 89 90 1 23 6 57

In Process 2: 45 64 23 14 23 56

List the contents of the array B in the process with rank 2 after executing the following routine.

MPI_Allgather(A,2,MPI_INT,B,2,MPI_INT,MPI_COMM_WORLD);

15 20 6 57 23 56 | 12 34 89 90 45 64 | All the values of Process0, then Process 1 and Process | 12 89 45 34 90 64

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

Q.No : 9) **Score : 0.00 / 0.50**

Values of integer array A is as shown below in 3 processes in which process 0 being the root:

In Process 0: d e f

In Process 1: 1 2 3

In Process 2: a b c

List the contents of the array B in the process with rank 2 after executing the following routine.

MPI_Alltoall(A,1,MPI_INT,B,1,MPI_INT,MPI_COMM_WORLD);

- d 1 a e 2 b f 3 c
- f 3 c
- d e f 1 2 3 a b c
- d 1 a

Q.No : 10) **Score : 0.50 / 0.50**

Compute the time taken in temporal parallelism to correct 1000 papers having 4 questions, if 4 teachers are employed to evaluate one answer and each answer requires 5 minutes to evaluate.

- 5015
- 5025
- 5000
- 20000

Q.No : 11) **Score : 3.50 / 4.00**

Develop a MPI program to read a word *Str* of length *wordsize* in the root process, where *wordsize* is evenly divisible by total *size* number of processes. Using point to point routines root sends equal data to all processes. Let each process (including root) check received characters and if there are digits extracts it. Store the sum of digits in variable *Sum* otherwise store a value 0. Print the extracted digits and partial sum in each process with rank. Using collective communication routine, find the total sum of all the digits extracted in each process and print total sum in root process.

Str: Pc3ap23mi12dexam Size : 4

Process 0: 3 Sum : 3 Process 1: 23 Sum : 5

^ Top

ENG US 1:15 pm 14/10/2023

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

Q.No : 11) **Score : 3.50 / 4.00**

Develop a MPI program to read a word *Str* of length *wordsize* in the root process, where *wordsize* is evenly divisible by total *size* number of processes. Using point to point routines root sends equal data to all processes. Let each process (including root) check received characters and if there are digits extracts it. Store the sum of digits in variable *Sum* otherwise store a value 0. Print the extracted digits and partial sum in each process with rank. Using collective communication routine, find the total sum of all the digits extracted in each process and print total sum in root process.

Str: Pc3ap23mi12dexam Size : 4

Process 0: 3 Sum : 3 Process 1: 23 Sum : 5

Process 2: 12 Sum : 3 Process 3: Sum : 0

Total Sum in process 0 : 11

Page:1

```
#include "mpi.h"
#include <string.h>
#include <stdio.h>
void main (int *argc, char *argv[])
{   MPI_Status &status;
    int rank, size;
    MPI_Init(&argc, &argv);
```

```
#include <stdio.h>
void main (int *argc, char *argv [])
{     MPI_Status &status;
      int rank, size;
      MPI_Init(&argc, &argv);
      MPI_Comm_rank(&rank, MPI_COMM_WORLD);
      MPI_Comm_world(&size, MPI_COMM_WORLD);
      int wordsize, chunk, i, len;
      char str[200], substr[200];
      int sum=0, sarray [size];
  if (rank == 0)
```

```
MPI_Comm_size(&size, MPI_COMM_WORLD);
    int wordsize, chunk, i, len;
    char str[200], substr[200];
    int sum=0, sarray[size];
if (rank==0)
  { printf("Enter the string :");
     scanf("%s", str);
     wordsize= strlen(str);
     chunk= wordsize/size;
  for(i=1; i<size; i++)
{ MPI_Send(&chunk, 1, MPI_INT, i, i, MPI_COMM_WORLD);
```

```
chunk = wordsize / size ;
for(i=1; i<size; i++)
{ MPI_Send(&chunk, 1, MPI_INT, i, i, MPI_COMM_WORLD);
  MPI_Send( str, chunk, MPI_CHAR, i, i+1, MPI_COMM_WORLD);
}
}

else if (rank!=0)
{ char num[100]; int k=0 ;
  MPI_Recv(&len, 1, MPI_INT, 0, rank, MPI_COMM_WORLD, &status);
```

Page:2

```
MPI_Recv(substr, len, MPI_CHAR, 0, rank+1, MPI_COMM_WORLD,
```

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D


```
MPI_Recv(substr, len, MPI_CHAR, 0, rank+1, MPI_COMM_WORLD,
                                                    &status);
}
for(i=0; i<len; i++)
{   if(substr[i]=='0' || substr[i]=='1' ||
       substr[i]=='2' || substr[i]=='3' ||
       substr[i]=='4' || substr[i]=='5' ||
       substr[i]=='6' || substr[i]=='7' ||
       substr[i]=='8' || substr[i]=='9')
    {  sum += int(substr[i])
       num[k]= substr[i];
         k++;
    }
```

^ Top

Search ENG US 1:16 pm 14/10/2023

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

```
num[k]= subst[i];
    k++;
  }
printf("Process %d: %s , Sum: %d", rank,
        num, sum);
}
MPI_Gather(&sum, 1, MPI_INT, Sarray, 1, MPI_INT,
        0, MPI_COMM_WORLD);
if (rank==0)
{   int TotSum= 0;
    for (i=0, i<size; i++)
```

^ Top

ENG US 1:16 pm 14/10/2023

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

```
{    int TotSum = 0;
     for (i=0; i< size; i++)
         TotSum += Sarray[i];
  Printf("Total Sum in process 0 : %d",
                 TotSum);
  }
 MPI_Finalize();
}
```

Q.No : 13) **Score : 2.00 / 3.00**

Design a Cuda application that converts an RGB image to a Grayscale image. Write only the kernel code.

Page:1

```
__global__ rgbToGrayScale(int *image, int *final, int n,
```

^ Top

Search ENG US 1:16 pm 14/10/2023

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

Q.No : 13) **Score : 2.00 / 3.00**

Design a Cuda application that converts an RGB image to a Grayscale image. Write only the kernel code.

Page:1

```
__global__ rgbToGrayScale(int *image, int *final, int n,
                          int m)
{
    int tid = (blockIdx.x * blockDim.x) + threadIdx.x;
    int rgb = threadIdx.x;
    int rgbChannel = rgb * CHANNELS;
    int r = image[rgbchannel];
    int g = image[rgbchannel] + 1;
    int b = image[rgbchannel] + 2; ;
```

Search ENG US 1:16 pm 14/10/2023

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

```
int rgb = threadIdx.x;
int rgbChannel = rgb * CHANNELS;
int r = image[rgbchannel];
int g = image[rgbchannel] + 1;
int b = image[rgbchannel] + 2; ;

final[tid] = r + g + b;
}
```

^ Top

Search ENG US 1:16 pm 14/10/2023

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

Q.No : 14) **Score : 3.00 / 3.00**

Design a Cuda application to add two vectors. Write the host and kernel code.

Page:1

```
#include "cuda.h"
#include <stdio.h>
__global__ void vecAdd(int *d_a, int *d_b,
        int *d_c, int n)
{   int tid= (blockIdx.x * blockDim.x) +
                 threadIdx.x;
    if( tid < n)
    {
     d_c[tid] = d_a[tid] + d_b[tid];
```

^ Top

Search ENG US 1:16 pm 14/10/2023

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

```
        }
            d_c[tid] = d_a[tid] + d_b[tid];
        }
}
int main()
{
    int n, i;
    int *a, *b, *c;
    int *d_a, *d_b, *d_c;
    printf("enter size of vectors: ");
    scanf("%d", &n);
size_t size = n * sizeof(int);
cudaMalloc((void**)&d_a, size);
```

^ Top

ENG US 1:16 pm 14/10/2023

```
size_t size = n * sizeof(int);
    cudaMalloc((void**)&d_a, size);
    cudaMalloc((void**)&d_b, size);
    cudaMalloc((void**)&d_c, size);
        printf("enter vector a:");
          for(i=0; i<n; i++)
              scanf("%d", &a[i]);
        printf("enter vector b:");
            for(i=0; i<n; i++)
```



```
scanf("%d", &b[i]);
```

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

Page:2

```
scanf("%d", &b[i]);

cudaMemcpy(d_a, a, size, cudaMemcpyHostToDevice);
cudaMemcpy(d_b, b, size, cudaMemcpyHostToDevice);

vecAdd<<<1,n>>>(d_a, d_b, d_c, n);

cudaMemcpy(c, d_c, size, cudaMemcpyDeviceToHost);
printf("final sum vector:\n");
for(i=0; i<n; i++)
```

^ Top

ENG US 1:16 pm 14/10/2023

```
cudaMemcpy(c, d_c, size, cudaMemcpyDeviceToHost);
printf("final sum vector:\n");
  for(i=0; i<n; i++)
    printf("%d\t", c[i]);
 cuda_Free(d_a);
 cuda_Free(d_b);
 cuda_Free(d_c);
  return 0;
}
```

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

Q.No : 15) **Score : 3.00 / 3.00**

Point out the Cuda qualifiers for kernels, clearly specifying where they are executed and where they are invoked from.

Page:1

1. __host__

These kernels can only run on the host. They are accessed in all host functions and can only be called from another host function.

A function without a qualifier is a __host__ kernel.

^ Top

Search ENG US 1:16 pm 14/10/2023

2. __Device__

These kernels are executed in device memory & can only be invoked in another device kernel. They can be called recursively.

3. __global__

These kernels are executed on the device but can be invoked in host and device programs

3. __global__

These kernels are executed on the device but can be invoked in host and device programs.

These can be a combination of __host__ and __device__ as well that causes the kernel to act similar to __global__ qualified kernel.

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

Q.No : 16) **Score : 2.50 / 3.00**

Values of integer array A is as shown below in 4 processes in which process 0 being the root:

A value in Process 0: 0, 17, 34, 51

A value in Process 1: 5, 22, 39, 56

A value in Process 2: 10, 27, 44, 61

A value in Process 3: 15, 32, 49, 66

Let B and C are 1D integer arrays initialized with a value 0.

i)Write an MPI function call with appropriate parameters to display the following output using value of A as shown above.

B value in Process 0: 0, 5,10,15

B value in Process 1: 17, 22,27,32

B value in Process 2: 34, 39,44,49

B value in Process 3: 51, 56,61,66

ii) Write an MPI function call with appropriate parameters to display the following output using value of A as shown above.

C value in Process 0: 0, 17, 34,0

C value in Process 1: 5, 39, 73,0

C value in Process 2: 15, 66, 117,0

C value in Process 3: 30, 98, 166,0

Page:1

i)

MPI_alltoall (A, 1, MPI_INT, B, 1, MPI_INT,

^ Top

ENG US 1:17 pm 14/10/2023

i)

MPI_alltoall(A, 1, MPI_INT, B, 1, MPI_INT,
MPI_COMM_WORLD);

each element in
ii) resultant C is a cumulative sum of vertically aligned elements of all A arrays, except the last value is all C arrays is zero.

```
for(i=0; i<=2; i++)
```

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

last value is all C arrays is zero.

```
for(i=0; i<=2; i++)
MPI_Scan(&A[i], &C[i], 1, MPI_INT, MPI_ADD,
                              MPI_COMM_WORLD);
C[3] = 0;
```

^ Top

Search ENG US 1:17 pm 14/10/2023

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

Q.No : 17) **Score : 3.00 / 3.00**

Compare data and temporal parallel processing (any six).

Page:1

Data Parallel Processing	Temporal parallel processing
1. All of the Jobs are divided into chunks and each Process works on a single chunk.	Each Job is divided into several tasks and each task is computed by a separate process.
2. Does not require Synchronization.	Requires synchronization between all working processes

^ Top

ENG US 1:17 pm 14/10/2023

2. Does not require synchronization.	Requires synchronization between all working processes
3. Is unaffected by bubbles	Bubbles cause delay in execution time.
4. time taken: $(n*q) + \frac{(n*p)}{k}$ Speed up: $k / 1 + (k^2 * q / n*p)$	Time taken: $\frac{P \times (k+n-1)}{k}$ Speed up: $k / 1 + ((k-1)(n))$

n: total jobs, p: time taken per task, k = no. of task/no. of divisions, q: time taken to divide 1 division.

5. Requires total number of jobs to be equally divisible by no. of processes	Requires every job to be independently divisible
6. Faults in a single process do not affect overall task execution, but will delay completion	Faults in individual processes affects the execution of program.

Student Answer Script View - Google Chrome

manipal.examcloud.in/reports/viewevaluation.php?flag=1&ev=MTcxMw%3D%3D&ex=NDgxMDA%3D&aid=MTcyMTcxNw%3D%3D&evlcycl=MQ%3D%3D

Q.No : 18) **Score : 2.00 / 2.00**

Illustrate the main disadvantages of data parallelism with dynamic assignment.

Page:1

1. Data parallelism with dynamic assignment may lead to formation of queues amongst the processes. This would lead to multiple process being idle during wait time in queue.
2. The head process that distributes jobs is idle for most of the execution time
3. If there is a fault in head process, it

^ Top

30°C Mostly sunny Search ENG US 1:17 pm 14/10/2023

2. The head process that distributes jobs is idle for most of the execution time

3. If there is a fault in head process, it affects all of the other working processes

4. Scalability is difficult as having a large number of process would cause increased queue lengths.