CS528-MidSem-Part-B

CS528-MidSem-Part-B

Points: 27/36

1

Roll No *

190101056

2

Name *

Mushanolla Pranathi

Incorrect 0/1 Points

3

Can a thread of process access stack area of other threads of the same process?

- It is not porssible as stack area of differnt threads are different
- It is possible as the whole process memory address is shared by all the threads
- We cannot say

Correct 1/1 Points

4

Given a system of N nodes and our aim is to interconnect them, and our goal is to minimize the diameter. Which interconnection network is preferable

- Star
- Hypercube
- Tree
- Clique/Complete Graph

✓ **Correct** 1/1 Points

5

Given the following kernel, the best way to speed up this in Modern day processors of a single machine (PC/Laptop)

```
unsigned char x[N], w[N], k; //N is large
for(i=0;i<N;i++)  x[i]=(w[i]*x[i])-k;
```

- OpenMP
- OpenMP + AVX ✓
- MPI
- Pthread

✓ **Correct** 2/2 Points

6

Given a cache of size 1MB with 4 way set associative and block (or line) size of 64B, calculate the number of bits required for the index field

12

✓ **Correct** 2/2 Points

7

Given Ppeak and bs values of a machine 10 GF/s and 12 GB/s respectively, what will be the achieved performance of the following loop in Giga flop per second. (assume write allocate)

```
float s, a[N], b[N];
                for(i=0;i<N;i++) s=s+a[i]*b[i];
```

3

✓ **Correct** 1/1 Points

8

Why matrix application is an excellent candidates for GPU acceleration?

- Matrix application require high computation per data
- Matrix application is data intensive application and it is cache friendly
- Matrix application is highly parallel
- All of the above ✓

✓ **Correct** 2/2 Points

9

Programming model for GPU: tick the wrong one

- Threads get scheduled to SPs in phasewise and Thread-blocks get scheduled to SMs in phase wise
- Threads get mapped to SPs/CudaCore/StreamCores
- All the SPs and SMs share the load of threads and blocks equally pre-emptively ✓
- Thread blocks gets scheduled to SMs.

✗ **Incorrect** 0/2 Points

10

Suppose there are 10 identical processors and LPT rule is used to schedule the independent tasks with arbitrary execution without pre-emption to minimize Cmax, the achievable approximation can be ___. (one word or numeric value)

3/2

Correct answers: 4/3-1/3*10 | 1.3

✓ **Correct** 2/2 Points

11

Given a system of N nodes with hypercube interconnection network. The system has a diameter, the total number of links, and bisection bandwidth

- 2*Sqrt N, 4N, Sqrt N
- Log N, N*LogN, N/2
- Sqrt N, 4NLogN, LogN
- log N, (N/2)*logN, N/2 ✓

✗ **Incorrect** 0/2 Points

12

Suppose you have a Royal Enfield Bullet bike and the bike have fuel consumption model P=200+20*f^3, where f is speed of bike between 0 to 1. You needs to go IITG-Panbazar 30km by road with in 1 hour of travel time. Assume the bike speed corresponds 0-1 mean 0KMPH-100KMPH. Your aim is to reach the destination with in time deadline and minimize the fuel consumption.

- Better to go at maximum 100KMPH speed and reach the destination early ✓
- Better to go at the required speed 30KMPH ✗
- Better to go at some other speed to reduce fuel consumption
- We cannot say.

✗ **Incorrect** 0/2 Points

13

Why Single precision Floating Point (SFP) number have accuracy issue when looking at bigger numbers?

- Same 32 bits is used for both SFP representation and integer represenation
- Integers are equally spaced, where as SFP numbers are not ✗
- SFP can represent both smaller number upto 2^-127 and bigger number upto 2^127.
- Because of all the above reasons ✓

✓ **Correct** 2/2 Points

14

Given an application with a serial fraction value 0.1 and the rest of parallel fraction is divisible load. Calculate the maximum achievable speed up even if we are using infinite number of processor. (numeric answer)

10

✓ **Correct** 1/1 Points

15

Choose the best option about the dynamic and static construction of array in C++

- Static creation of the array is always beneficial as it save time
- Always go for dynamic array creation of the array as it save space
- If the probability of requirement of the array is low create lazily otherwise construct statically. ✓
- We cannot say.

✗ **Incorrect** 0/2 Points

16

What is the best possible optimization, we can think of the following code [hint: code don't take any external input]

```
int X=0, Y=0; N=1000;
        for (i=1;i<=N;i++) X=X+sin(i%5);
        for (i=1;i<=N;i++) Y=X+cos(i*i %5);
        printf("X=%d, Y=%d",X,Y);
```

- Merge both the for loop
- Use AVX and Simidization ✗
- Use copy propagation and static calculation ✓
- using LUTs for cos and sin computation

✓ **Correct** 1/1 Points

17

Calculate the load factor, dilation, and congestion for the embedding of 16 nodes m esh onto 4 node mesh.

- 4, 1, 2 ✓
- 4, 2, 1
- 4, 1, 1
- 2, 1, 2

✓ **Correct** 1/1 Points

18

Is this loop is beneficial to use GPU acceleration?

```
void VectorAvg(){
        for(int j=1;j<N-1;j++)
         A[j]=(B[j-1]+B[j]+B[j+1])/3.0;
        }
```

- Yes
- No ✓
- Maybe
- may not be

✓ **Correct** 2/2 Points

19

Calculate Span of the following DAG, assume execution time of A, B, C, D, E, F, G as 1, 2, 3, 4, 5, 6, and 7.

Prec: A->B, B->C, B->D, B->E, G->D, C->E, D->E, E->F

22

✓ **Correct** 1/1 Points

20

How can we simulate to multiprocessing on machine with one processor

- It is not possible to simulate multiprocessing on a single processor
- It is possible using time division multiplexing, many process get time share the processor to simulate the multiprocessing ✓
- It is possible to simulate at most two processes multiprocessing on single processor but not above 2 processes
- we cannot say

✓ **Correct** 2/2 Points

21

Choose the right explanation of the problem R |di, ri, pj | ΣUj

- Minimizing the number of missed tasks for tasks with infinite deadlines, release time, arbitrary execution time, pre-emption allowed on unrelated processor
- Minimizing the number of missed tasks for tasks with deadlines, release time, arbitrary execution time and pre-emption not allowed on unrelated processors ✓
- Minimizing the number of missed tasks for tasks with deadlines, online tasks, arbitrary execution time, pre-emption not allowed on uniform processors
- Minimizing number of missed tasks for tasks with deadlines, release time, arbitrary execution time, pre-emption allowed on uniform processors

✓ **Correct** 1/1 Points

22

Tick the correct statement about loop parallelization using simidization process of the given loop.

```
for (i=0; i<N; i++){
  p=3*i+4; q=6*i+2; r=2*i*i+25;
  X[p]=X[q]+r;
}
```

- Parallelization is definitely possible and there is no dependency between iteration ✓
- Parallelization is may be possible and there may be not having dependency between iterations
- Parallelization is not possible and there is a dependency between iteration
- We cannot say

✓ **Correct** 2/2 Points

23

Calculate total number of memory access in this code : *a memory access can be cache hit/miss*

```
int sum, data[2000] *dest=&sum;
for (i= 0; i< 2000; i++) *dest += data[i];
```

6000

✓ **Correct** 1/1 Points

24

Among these networks, which network has the highest bisection bandwidth

- Tree
- 2D Torus ✓
- Ring
- Star

✓ **Correct** 2/2 Points

25

Suppose you have a Hero Honda Spendour bike and the bike have fuel consumption model P=5+50*f^3, where f is speed of bike between 0 to 1. You needs to go IITG-Panbazar 30km by road within one hour of travel time. Assume Hero Honda Spendour bike speed corresponds 0-1 mean 0KMPH-100KMPH. Your aim is reach the destination within deadline travel time and minimize the fuel consumption.

- ○ Better to go at maximum 100KMPH speed and reach the destination
- ◉ Better to go at the required speed 30KMPH ✓
- ○ Better to go at some other speed to reduce fuel consumption
- ○ We cannot say.