SET-X

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJ.)
II SEMESTER 2021-2022
Mid Semester Test - PART A (CLOSED BOOK)

Course No.: CS F422 **Course Title:** Parallel Computing
Date: 11th March (9-10:30) **Maximum Marks:** 10% (10M) (35 Mins)

Note:
- Write answers in this sheet itself.
- Overwritten answers will not be accepted for rechecks
- Once you submit PART-A, you can collect PART-B.

IdNo: _____ **Name:** _____

Q1. Write answers to the following questions in the grid given below. A question may have more than one correct option. Marks will be awarded only if all correct options and only correct options are chosen. [10*0.5=5M]

1	2	3	4	5	6	7	8	9	10

1) Speedup of a given program with 0.25 fraction of serial execution on 16 processors is ___.
2) Hardware speculative execution is
   (a) Static ILP
   (b) Dynamic ILP
   (c) scheduling instructions within a control dependency
   (d) scheduling instructions across control dependencies
3) Pipelining in ILP
   (a) Increases instruction latency
   (b) Increases instruction throughput
   (c) Decreases instruction latency
   (d) Decreases instruction throughput
4) The loop $for(i = 0; i < N; i+= 10)\ A[i] *= 3;$ has
   (a) Good data temporal locality
   (b) Good data spatial locality
   (c) Loop carried dependence
   (d) No loop carried dependence
5) Snooping technique for cache coherence
   (a) Needs shared bus
   (b) Needs a directory
   (c) Works with write-update
   (d) Works with write-invalidate
6) Scaled speedup of a given program with 0.3 fraction of serial execution on 20 processors is ___.
7) Scalable parallel system is
   (a) One which has increased speedup with respect to increased number of processors
   (b) One which has increased efficiency with increasing processors
   (c) One which maintains efficiency despite increasing processors and work
   (d) One which has same cost despite increasing processors
8) In PRAM model, time taken to spawn p processes is _____.
9) Efficiency of a parallel program is defined in terms of Speedup (S) and processors (p) as ___.
10) Parallel algorithm A is cost-optimal if cost of A same as that of _____.

1. Write **brief** answers. [5M].

2. Given a parallel algorithm A with computation time *t*, if A performs *m* computational operations, one can construct an algorithm A' to perform the same work with *p* processors in time _____. Justify. [0.75M]

3. List the PRAM models (CRCW (Common, Priority), EREW, CREW) from the strongest model to the weakest model? [0.75]

4. Derive time complexity of parallel reduction algorithm using PRAM model with *n/2* processors on *n* elements. [0.75]

5. What is the time complexity and efficiency of a parallel algorithm for merging two sorted lists of n/2 elements each on n processors using PRAM model? [0.75]

6. Write a code segment using OpenMP directives to add 10000 numbers using 10 processors. [2M]

-&--

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJ.)
II SEMESTER 2021-2022
Mid Semester Test - PART B (OPEN BOOK)

Course No.: CS F422 **Course Title:** Parallel Computing
Date: 11th March (9-10:30) **Maximum Marks:** 15% (15M)

Note:
- Overwritten answers will not be accepted for rechecks
- Write all parts of a question together.

Q1. Briefly answer the following.

(a) False sharing will not happen if all processors in a shared memory system use different bytes in a single cache line. State true or false with justification.

(b) All loops can be unrolled by the compiler to increase ILP. Give an example to counter this.

(c) Explain why log barrier complexity is better than linear barrier in phreads?

(d) In which case, guided scheduling class performs better than dynamic scheduling class in OpenMP loop parallelization. Give an example.

[4*1.5=6M]

Q2. Consider the following expression to compute e^x.

$$1+\frac{x}{1!}+\frac{x^2}{2!}+\frac{x^3}{3!}+\frac{x^4}{4!}+\ldots\ldots\ldots$$

Design and write a program using Pthreads to compute e^x for N terms and T threads where N is taken on command line and T is number of threads. N>>T. [5M]

Q3. Consider the following program. Using OpenMP directives, make it parallel. [4M]

```
int main()
{
    int i, n, sum=0;
    /* Input upper limit from user */
    printf("Enter upper limit: ");
    scanf("%d", &n);

    for(i=2; i<=n; i+=2)
    {
    /* Add current even number to sum */
        sum += i;
    }
    printf("Sum of all even number
between 1 to %d = %d", n, sum);
    return 0;
}
```

--&--