Exam Date & Time: 20-Jun-2024 (02:30 PM - 05:30 PM)

MANIPAL ACADEMY OF HIGHER EDUCATION

VI Semester, B.Tech. , Data Science and Engneering

PARALLEL PROGRAMMING [DSE 3254]

Marks: 50 **Duration: 180 mins.**

Makeup Examination, June 2024

Answer all the questions. Section Duration: 180 mins

Answer all Questions
Missing values may be suitably assumed

1) Compare and contrast data-level parallelism and task-level parallelism in OpenMP with appropriate examples. (4)

2) Evaluate the methods employed in OpenMP to localize global variables to each thread, and assess the significance of the copyin clause within this context. (3)

3) Compare and contrast with appropriate examples, MPI_Ssend and MPI_Bsend in point-to-point communication of MPI. (3)

4) Design a MPI program to perform parallel transposition of a matrix. The matrix is initially stored in the root process. The root process distributes portions of the matrix to different processes using point-to-point communication. Each process transposes its part of the matrix, and then the root process gathers the transposed parts and prints the final transposed matrix. (4)

5) Analyze the role of MPI_Bcast in collective communication, and demonstrate its emulation through the utilization of point-to-point communications in MPI (3)

6) Examine the memory hierarchy of CUDA thoroughly, and provide a detailed analysis of CUDA variable type qualifiers. (3)

7) Assume you are working on a physics simulation that involves calculating the potential energy of a large number of particles (N) in a closed environment. Each particle has a mass and height associated with it which are stored in two different arrays M and H. Each particle's potential energy is calculated as P=m*G*h where m is the mass of the particle and h is the height of the particle. Write a CUDA program that efficiently calculates the potential energy of all the individual particles in parallel and stores it in an array Q. Also, write a reduction algorithm to estimate the particle with minimum potential energy from Q. Assume that, N=1200 and a block can have 256 threads. (5)

8) Consider the following statement: “Shared memory has lower latency and higher bandwidth than registers”. State true or false and justify your answer. (3)

9) Discuss the concept of “zero-overhead thread scheduling” in the context of CUDA. (2)

10) Discuss the concept of thread divergence, focusing on its implications in solving the reduction problem. Provide an in-depth explanation and illustrate it with a CUDA program designed to calculate the sum of elements in an array. (4)

11) Assess the rationale behind avoiding the placement of parallel regions within inner loops in OpenMP. (3)

12) Evaluate and illustrate the application of "*loop interchange*" and "*loop tiling*" techniques with suitable examples. (3)

13) Assume you are working on a real-time gray-scale image processing application. The processing involves a pre-processing step where you are supposed to determine the transpose of the given input image. Further, you are tasked to find the sum of all the pixels in every row of the transposed image and store the results in a separate vector G. Design a CUDA program to do the above tasks parallelly. Assume that the image dimension is 780x650. Also, consider that the streaming multi-processor can take a maximum of up to 2048 threads and it allows up to 1024 threads in each block. Assume that the Image is already read into variable I and the size of the image is already estimated and stored in variable SIZE_I. (5)

14) Discuss the concept of a "performance cliff" in CUDA. Provide a detailed discussion into the challenges associated with inefficient memory accesses in CUDA programming. (3)

15) Given an original program with a serial execution time of 9.4 seconds, where 68% of its execution time has been parallelized, and employing 6 processors introduces a total overhead of 0.24 units to the total CPU time, calculate the speedup and efficiency achieved by the parallel program with 6 processors. Additionally, estimate the total CPU time and elapsed time of the program. (2)

-----End-----