S.P.P.U. External Practical Viva Questions and Answers

Name: HK

Viva Questions and Answers (Parallel Bubble Sort using OpenMP)

1. What is the objective of your project?

Answer:

The objective is to implement and parallelize the bubble sort algorithm using OpenMP to improve sorting performance by utilizing multiple CPU cores.

2. What is OpenMP?

Answer:

OpenMP (Open Multi-Processing) is an API that supports multi-platform shared memory multiprocessing programming in C, C++, and Fortran.

3. Which libraries have you used and why?

Answer:

- iostream - for input/output operations.
- stdlib.h - standard library for dynamic memory allocation.
- omp.h - OpenMP header file for parallel programming constructs.

4. How is the bubble sort parallelized in your program?

Answer:

The algorithm alternates between even and odd indexed passes, and uses #pragma omp parallel

for to perform independent comparisons and swaps concurrently within a pass.

5. What is the role of 'first = i % 2' in the code?

Answer:

It determines whether the current pass should compare even or odd indexed pairs, ensuring that adjacent elements are compared properly in parallel.

6. What does '#pragma omp parallel for shared(a,first)' do?

Answer:

It tells the compiler to execute the 'for' loop in parallel, sharing the array 'a' and the variable 'first' among all threads to avoid data conflicts.

7. Why is the swap operation important here?

Answer:

Swapping adjacent elements ensures that larger elements move toward the end, which is the basic principle of bubble sort.

8. How is dynamic memory allocated in this program?

Answer:

Dynamic memory is allocated using 'new int[n]', allowing flexibility based on user input size.

9. What are the potential problems when parallelizing bubble sort?

Answer:

Data dependency: simultaneous swaps might interfere if adjacent elements are modified incorrectly. The even-odd approach prevents these conflicts.

10. How can you improve the performance further?

Answer:

By minimizing synchronization overhead, optimizing thread scheduling, or using more advanced parallel sorting algorithms like parallel merge sort.

Best of Luck, HK!