Introduction to OpenMP for Parallel Programming

Course Objectives - This course aims not only to teach OpenMP but also to foster a deeper understanding of parallel programming concepts**.** In this process, learners will be aware of parallel programming and understand the different principles on which one stands. It is to also teach learners how to use OpenMP for parallelizing C/C++ and Python code. Furthermore, it is about developing skills in the learners to analyze and optimize parallel applications.

Course Outline

I. **Introduction to Parallel Programming –** Give an Overview of Parallel Computing, the different Types of parallelism (data, task), some Applications of parallel computing, its application in various fields.

Introduce OpenMP- Tell us what it is, its history and evolution and some OpenMP architecture and components.

II. **OpenMP Basics –** covers OpenMP Directives**.** Understanding directives, runtime library routines, and environment variables**,** some syntax and structure of OpenMP programs

As for in-Class activity: Teach the class how to set up the environment (for exple in visual studio). Then, write one or two simple OpenMP programs using #pragma omp parallel that you will be compiling and running OpenMP Programs. In this process, pay some attention to compilation flags and execution

III. **Work Sharing Constructs** – represent a crucial topic in parallel computing with OpenMP. Thus, it will be more than important to cover some work sharing directives such as #pragma omp for, #pragma omp sections, #pragma omp single, #pragma omp master … among many others, load balancing and performance considerations.

As for in-Class Activity: Take a serial program of your choice and modify it to use work-sharing constructs

IV. **Synchronization -** A very important topic to cover. In this process, it will be nice to make us understand the concepts of race conditions and data hazards, OpenMP synchronization constructs (critical, atomic, barrier…..)

As for in-Class Activity: Implement synchronization in a parallel program of your choice.

V. **Advanced OpenMP Features**

Introduction to tasks and task scheduling using #pragma omp task

As for in-Class Activity: Create a program that utilizes tasking for dynamic workload distribution

Nested Parallelism should be covered.

As for in-Class Activity: Experiment with nested parallel regions in a sample application. In this process, one should be able to enable and use nested parallel regions

VI. **Performance Analysis –** This is mainly about profiling, identifying bottlenecks and performance tuning. For this, tools for profiling OpenMp applications (e.g gprof, Intel Vtune,…) should be discussed

As for In-Class Activity: Use profiling tools on existing OpenMP programs. In this, you can talk about techniques for optimizing parallel programs, and Memory access patterns and cache optimization

As for In-Class Activity: Optimize a given OpenMP program for better performance.

Textbooks/Materials:

- "Using OpenMP: Portable Shared Memory Parallel Programming" by Barbara Chapman et al.
- Online documentation and resources from the OpenMP website.
- Introduction to OpenMP: 11 part 1 Module 6

Software:

- GCC or Intel compilers with OpenMP support.
- Profiling tools as mentioned above.