UCS645: Parallel and Distributed Computing

LAB 1: Introduction to OpenMP

Aim

The aim of this experiment is to understand the basics of OpenMP and shared memory parallel programming by implementing and analyzing parallel programs using C and OpenMP.

Software Requirements

1. Linux OS / WSL
2. GCC Compiler with OpenMP support
3. Basic knowledge of C programming

Experiment 1: DAXPY Loop

The DAXPY operation performs the computation X[i] = a*X[i] + Y[i] on vectors of size 2^16. The experiment was performed by varying the number of threads starting from 2. Execution time was recorded to observe speedup.
Observation: Speedup increases with number of threads until the number of physical cores is reached. Beyond this, performance saturates or degrades due to overhead and context switching.

Experiment 2: Parallel Matrix Multiplication

Matrix multiplication of size 1000x1000 was implemented using OpenMP. Two parallel approaches were used: 1D parallelization using a single loop and 2D parallelization using nested loops with collapse clause.
Observation: 2D parallelization provides better load balancing and improved performance compared to 1D parallelization due to effective utilization of threads.

Experiment 3: Calculation of π

The value of π was approximated using numerical integration and parallelized using OpenMP reduction to avoid race conditions.
Observation: Reduction clause ensures correctness and significantly improves performance with increasing number of threads.

Conclusion

This lab demonstrated the effectiveness of OpenMP for shared memory parallel programming. Proper use of directives such as parallel, for, reduction, and scheduling leads to improved performance while maintaining correctness.