Maithreepala U.G.R.D.

EG/2020/4062

Project Proposal : EC7207 - High Performance Computing

Title: Parallel Performance Evaluation of Sparse Matrix-Vector Multiplication (SpMV)

Introduction

Sparse Matrix-Vector Multiplication (SpMV) is a fundamental operation in various scientific and engineering fields, including finite element methods, machine learning, and graph analytics. These applications often involve very large and sparse matrices, where most elements are zeros. Efficient storage and computation strategies are essential for optimizing memory usage and processing time. This project explores the parallel implementation of SpMV using multiple parallel programming paradigms—OpenMP for shared memory systems, MPI for distributed memory systems, and a hybrid MPI + OpenMP model. Through this, we aim to evaluate and compare the performance benefits and limitations of each model in terms of speedup, scalability, and efficiency.

Objectives

- Implement Sparse Matrix-Vector Multiplication using a suitable sparse format (e.g., CSR - Compressed Sparse Row).
- Evaluate and compare the performance of the following approaches:
  - Sequential (single-threaded) implementation
  - OpenMP-based shared memory parallelization
  - MPI-based distributed memory parallelization
  - Hybrid MPI + OpenMP parallelization
- Conduct performance analysis by measuring execution time and calculating speedups for matrices of varying sizes and sparsity levels.
- Study the scalability of each approach by running experiments on multiple cores/nodes.
- Identify bottlenecks and analyze the trade-offs between the different parallelization techniques.

Expected Outcomes

- Optimized and verified implementations of SpMV using sequential, OpenMP, MPI, and hybrid MPI + OpenMP approaches.
- Comparative performance analysis showing:
  - Execution times
  - Speedup curves
  - Scalability plots for each implementation
- Insights into how matrix structure (size and sparsity) impact the effectiveness of each parallel model.
- A final report summarizing implementation details, performance results, and recommendations for suitable parallel models based on application context and system architecture.