SLIIT
Discover Your Future

BSc (Hons) in Information Technology
Year 3

Assignment 03

SE3082 – Parallel Computing **Semester 1, 2025**

Introduction

You are required to identify a suitable algorithm for parallelization from one of the recommended problem domains listed below. These domains represent areas where parallel computing can provide significant performance improvements. You are expected to find or develop a serial C implementation of an algorithm within your chosen domain and then create parallel versions using different technologies.

You need to get approval for the algorithm that you are planning to parallelize before proceeding with the implementation phase.

Recommended Problem Domains for Parallel Algorithms

Students should select an algorithm from ONE of the following domains:

1. Image Processing and Computer Vision

- Image filtering (Gaussian blur, edge detection, sharpening)
- Image transformations (rotation, scaling, convolution)
- Histogram equalization
- Image segmentation
- Feature detection algorithms

2. Numerical Computation and Scientific Computing

- Matrix operations (multiplication, inversion, decomposition)
- Numerical integration methods (Trapezoidal rule, Simpson's rule)
- Solving systems of linear equations
- Monte Carlo simulations
- N-body simulation problems

3. Sorting and Searching Algorithms

- Advanced sorting algorithms (Merge sort, Quick sort, Bitonic sort)
- Parallel search algorithms
- String matching algorithms
- Graph traversal algorithms (BFS, DFS)

4. Cryptography and Security

- Encryption/Decryption algorithms (AES, DES)
- Hash function computations
- Brute force password cracking simulations
- Prime number generation and testing
- Digital signature verification

5. Data Mining and Machine Learning

- K-means clustering
- K-Nearest Neighbors (KNN)
- Decision tree training
- Neural network operations (forward/backward propagation)
- Data preprocessing operations (normalization, feature scaling)

Submission Requirements

Phase 1: Algorithm Approval – Deadline 5th of November 2025

Submit your proposed algorithm by sending an email to **[nuwan.k@sliit.lk]** with the subject line **"SE3082 – Assignment Proposal"**. Your email must include:

a) **Title of the Algorithm** you have chosen

b) **Problem Domain** (from the list above)

c) **Brief Description** of the algorithm (200-300 words explaining what the algorithm does and why it is suitable for parallelization)

d) **Serial C Code** or detailed pseudo code of the serial implementation (you can get this from the internet or generate a code using AI, ensure this code works)

- You may find existing implementations online or write your own or use AI
- Clearly cite any sources used

Note: Approval will be granted via email reply within 2 working days.

BSc (Hons) in Information Technology
Year 3

Assignment 03

SE3082 – Parallel Computing — Semester 1, 2025

Phase 2: Implementation and Evaluation

Part A: Parallel Implementations (60 marks)

You need to develop parallel versions of your approved algorithm using the following three technologies:

1. **OpenMP** (20 marks)
2. **MPI** (20 marks)
3. **CUDA** (20 marks)

Marking Criteria for Each Implementation:

- **Correctness and Functionality** (8 marks): The parallel implementation produces correct results
- **Parallelization Strategy** (7 marks): Effective use of the parallel programming paradigm and optimization techniques
- **Code Quality and Originality** (5 marks): Clean, well-commented code that shows original work (not direct copy-paste from internet sources)

Important: Direct copies of solutions found on the internet will receive **0 marks**. Your implementation should demonstrate your own understanding and approach to parallelization.

Part B: Performance Evaluation (25 marks)

Conduct comprehensive performance evaluation of all three implementations:

1. OpenMP Evaluation (6 marks)

- Vary the number of threads (e.g., 1, 2, 4, 8, 16)
- Provide graphs:
  - Number of threads vs Execution time
  - Number of threads vs Speedup
- Include screenshots showing execution with different thread counts

2. MPI Evaluation (6 marks)

- Vary the number of processes/nodes (e.g., 1, 2, 4, 8, 16)
- Provide graphs:
  - Number of processes vs Execution time
  - Number of processes vs Speedup
- Include screenshots showing execution with different process counts

3. CUDA Evaluation (6 marks)

- Vary the block size and number of threads per block
- Provide graphs:
  - Configuration parameters vs Execution time
  - Configuration parameters vs Speedup
- Include screenshots showing execution with different CUDA configurations

4. Comparative Analysis (7 marks)

- Compare all three implementations on the same dataset/problem size
- Create comparative graphs (execution time and speedup across all three)
- Justify which implementation would be most appropriate if sufficient computational resources are available
- Discuss the strengths and weaknesses of each approach for your specific algorithm

---

Part C: Documentation and Analysis (15 marks)

Submit a written report (3-4 pages, single-spaced, excluding images and graphs) covering:

1. Parallelization Strategies (4 marks)

- Detailed explanation of the parallelization approach used for each implementation
- Justification for design decisions made
- Load balancing and data distribution strategies

2. Runtime Configurations (3 marks)

- Hardware specifications used for testing
- Software environment details (compilers, libraries, versions)
- Configuration parameters for each implementation

3. Performance Analysis (4 marks)

- Analysis of speedup and efficiency metrics
- Identification of performance bottlenecks
- Discussion of scalability limitations
- Overhead analysis for each implementation

4. Critical Reflection (4 marks)

- Challenges encountered during implementation
- Limitations that restrict scalability
- Potential optimizations for future improvements
- Lessons learned about parallel programming paradigms

Submission Guidelines

Upload a **single ZIP file** to the link that will be provided, with a video recording of the 3 implementation been executed.

1. **Source Code Folder**
   - Separate subdirectories for Serial, OpenMP, MPI, and CUDA implementations
   - Include Makefiles or compilation instructions
   - README file with compilation and execution instructions
2. **Screenshots Folder**
   - Screenshots of all executions showing runtime configurations
   - Output verification screenshots
   - Performance monitoring screenshots (if applicable)
3. **Report Document** (PDF format)
   - Written analysis as specified in Part C
   - All graphs and performance evaluation results
   - References cited in IEEE format
4. **Data Files** (if applicable)
   - Input data files used for testing
   - Output files for verification

Academic Integrity

- You are expected to write your own code, use of AI is permitted (but must be cited, with prompts). While you may reference online resources for understanding concepts, direct copying of code from a source will result in zero marks.
- Properly cite all sources, references, and any code snippets adapted from other sources.
- Collaboration is not permitted. This is an individual assignment.
- Plagiarism will be dealt with according to university policies.