Parallel Data Analytics Across Architectures

In this project, you will select a real-world dataset of your choice (e.g., traffic patterns, stock data, weather trends, image sets, or sensor data) and design a parallel data analysis pipeline that explores and processes this dataset using four programming paradigms:

1. Sequential
2. Pthreads (CPU-level threads)
3. OpenMP (shared-memory parallelism)
4. MPI (distributed-memory parallelism)
5. CUDA (GPU parallelism)

The goal is to analyze the same dataset using each paradigm and compare performance, scalability, and efficiency across architectures.

Requirements:

1. **Dataset Selection:**
    - You may choose any dataset that is large enough to show measurable parallel performance.
2. **Parallel Implementations:**
    - Implement a computationally intensive algorithm on the dataset using:
        - sequential
        - pthreads
        - OpenMP
        - MPI
        - CUDA
3. **Performance Evaluation:**
    - Compare runtime, scalability, and speedup across the four implementations.
    - Test pthreads, OpenMP and MPI with at least 3 varying configurations (such as varying number of threads and processes).
    - For CUDA, consider the following:
        - Kernel Launch Configurations: Optimize thread block size, grid size, and occupancy.
        - Shared Memory Utilization: Reduce global memory access latency by leveraging shared memory.
        - Tiling Techniques: Apply tiling for memory-efficient computations.
        - Memory Coalescing: Optimize global memory access patterns to minimize latency.
4. **Deliverables:**
    - Code files (with results)
    - Short report (~5 pages) explaining:
        - Chosen dataset and operation
        - Algorithm or Pseudocode applied
        - Implementation details for each paradigm

        - Experimental setup (varying configurations and CUDA optimization)
        - Comparison table in terms of runtime, scalability, and speedup with discussion and analysis.
        - Project management: role of each team member in the project.
    - Deliver a 10-minute presentation (last week of classes) summarizing key findings and demonstrating the performance improvements.

5. **Evaluation Criteria:**
    - Project Selection & Application Complexity 5%
    - Algorithm 10%
    - Correctness of the 4 Implementations 40%
    - Testing on varying configurations 15%
    - Optimization Strategies for CUDA 15%
    - Performance Comparison and discussions 10%
    - Insights/Conclusion; Team Management: 5%

Only 1 submission per group please!

Important Notes

- Projects must show original work and distinct datasets or approaches per group.
- The codes should not be present online on github or other platforms. The work should be original. Otherwise, you get a zero on the project and you lose 20% of your final grade.
- Collaboration within the group is encouraged, but cross-group collaboration is not allowed.