COL380 A3 Report

Koduru Suchith (2021CS10572)

1 CUDA Implementation

The assignment focuses solely on the CUDA implementation. Below are the key optimizations incorporated:

1.1 Optimization Strategies

- **Memory Optimization**: Used shared memory to reduce global memory access latency.
- **Parallel Execution**: Launched multiple CUDA threads to distribute computations efficiently across GPU cores.
- **Coalesced Memory Access**: Ensured memory accesses are aligned to minimize memory transaction overhead.
- **Thread Synchronization**: Used `__syncthreads()` where necessary to avoid race conditions and ensure correct data dependency handling.
- **Kernel Optimization**: The kernel was designed to maximize parallel throughput while minimizing divergence in thread execution paths.

2 Performance Evaluation

Performance analysis was conducted by measuring execution time for different problem sizes. The CUDA implementation demonstrated significant speedup compared to sequential execution methods, particularly for larger input sizes.

3 Results and Observations

Experimental results indicate that the GPU implementation achieves notable speed improvements due to efficient parallel execution and memory optimizations.

4 Conclusion

The CUDA-based approach effectively optimizes the computation process by utilizing GPU parallelism. Further improvements can be explored by fine-tuning kernel configurations and leveraging advanced optimization techniques such as warp-level programming and asynchronous memory operations.