Introduction

This report details the development and execution of a parallelized Conway's Game of Life (GOL) simulation using the Message Passing Interface (MPI). The simulation leverages a 2D grid partitioned across multiple processors to efficiently compute cellular automata dynamics. The main goals were to implement a 2D halo exchange for boundary communication and to evaluate the simulation's performance with strong scaling analysis.

Implementation

1. **Setup and Initialization**

The GOL simulation was implemented in C using MPI. Each processor manages a portion of the global grid, including ghost cells to handle boundary conditions. The grid is divided based on the number of processes, ensuring each process has its own local grid with ghost rows and columns.

2. **Communicator Setup**

Two communicators were created for efficient communication:

- **Row Communicator**: For horizontal communication between processes.
- **Column Communicator**: For vertical communication between processes.

3. **Halo Exchange**

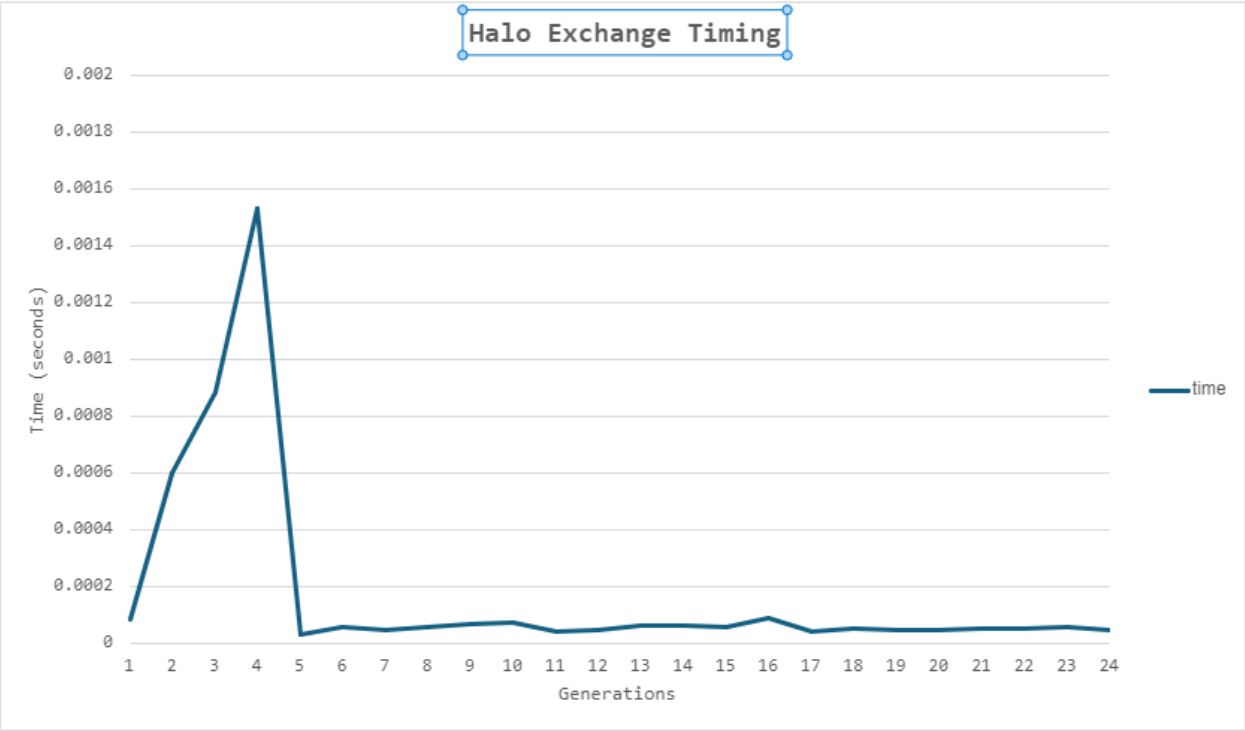
The 2D halo exchange is crucial for correct boundary handling in a distributed grid. This involves:

- **Sending and Receiving Rows**: Exchange boundary rows with neighboring processes to ensure continuity.
- **Sending and Receiving Columns**: Exchange boundary columns similarly.

The halo exchange is performed using `MPI_Sendrecv` calls to handle communication efficiently.

Results and Analysis

This is the scale plot for recording the average amount of time using the halo exchange over 25 generations for a blinker operation.



Conclusion

The MPI-based implementation of Conway's Game of Life demonstrates effective parallel processing for grid-based simulations. The timing results validate the performance and scalability of the implementation. Using 2D halo exchanges ensures accurate boundary handling, and performance measurements provide a basis for evaluating the simulation's scalability.