Assignment 1 Technical Document

The technical idea for this MPI program is to parallelize the computation of row sums of a 100x100 matrix across two processes to reduce overall computation time. Here's a detailed description of the approach:

Initialization

- The MPI environment is initialized at the beginning of the program using `MPI_Init`.
- Each process in the MPI program is assigned a unique rank, obtained using `MPI_Comm_rank`.
- The total number of processes is determined using `MPI_Comm_size`.

Data Distribution

- The matrix is divided between the two processes. Process 0 is responsible for rows 0-49, and Process 1 is responsible for rows 50-99.
- Process 0 generates the entire matrix but only sends the first half (rows 0-49) to Process 1 using non-blocking communication (`MPI_Isend`)

Parallel Computation

- Both processes compute the sum of their respective rows independently.
- Process 1 starts computing the row sums for the first 25 rows of its assigned range immediately after initiating the non-blocking receive operation to overlap communication with computation.

Communication

- Process 1 uses `MPI_Test` to check the completion of the non-blocking receive operation. Once the data is available, it completes the computation for the

Finalization

- The final row sums are printed by Process 0.
- The MPI environment is finalized using `MPI_Finalize`.

By distributing the data and computation across two processes and overlapping communication with computation, this MPI program efficiently utilizes parallel processing to reduce the execution time for calculating row sums of a matrix.