CM30225 Parallel Computing
Assessed Courseork Assignment 1

December 11, 2016

1 Approach

In order to parallalise the relaxation problem we need to be able to split the matrix up into chunks then let each node relax its own chunk with as little communication between nodes as possible.

In order to chunk the matrix up we want to give each node as similar numbers of rows as possible to distribute work evenly. In my implementation first we calculate the rounded up value of the number of rows minus 2, because the border rows arn't relaxed, divided by the number of nodes. Each node is the allocated this many rows until there are no rows left in the matrix. For example if we run 5 nodes on a 100 x 100 matrix the rounded value is 20 so the first 4 nodes are allocated 20 and the last is allocated 18.

The least each node needs to know after each iteration is the values in the row to the left and right of its chunk. For example given a 6 x 6 matrix and 2 nodes, the first is allocated row 2 and 3 and the second rows 4 and 5, after one iteration all the first nodes needs is row 4 and the second node only needs row 3. Therefore after each iteration each node sends its outside rows to the appropriate nodes.

We can improve this by calculating the two outside rows before any others and sending them, so the node can be calculating the rest of the chunk while the communication happens. To do this MPI_Isend and MPI_Irecv are used as these are non blocking so the node won't wait for the other node to receive the value before continuing. The psudocode for this is show below.

2 Parralisation Technique

3 Avoiding Race Conditions

4 Scalability Investigation

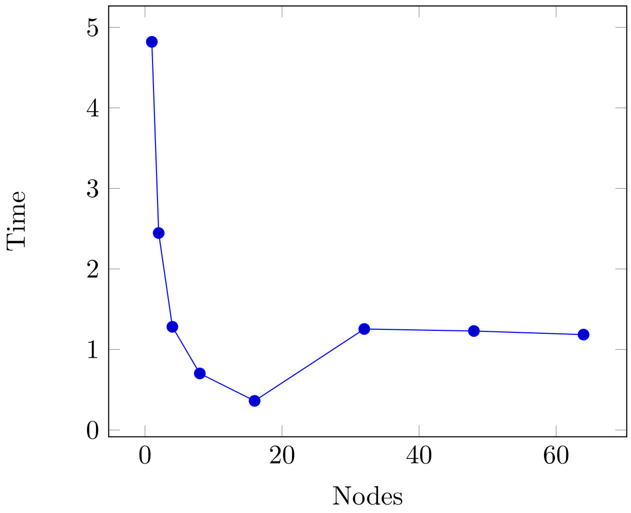

Figure 1:

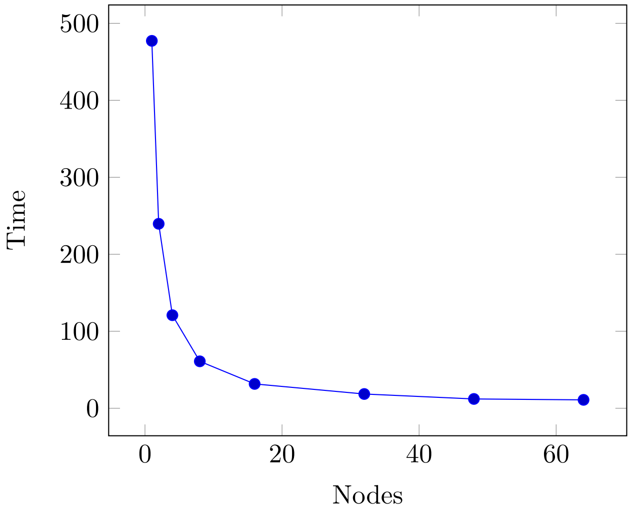

Figure 2: