1 Parallel Summation

For the parallel summation of \( N^2 \) integer numbers a 2-dimensional torus of size \( N \times N \) is used, where each computing node stores exactly one integer \( val \leftarrow rank+1 \). The parallel summation algorithm works as follows:

\[
\begin{align*}
\text{sum} & \leftarrow \text{val} \\
\text{tmp} & \leftarrow \text{val} \\
\text{for } i & \leftarrow 1 \text{ to } N-1 \text{ do} \\
& \quad \text{MPI\_send(tmp, 1, MPI\_INT, left neighbour, ...)} \\
& \quad \text{MPI\_recv(tmp, 1, MPI\_INT, right neighbour, ...)} \\
& \quad \text{sum} \leftarrow \text{sum} + \text{tmp} \\
& \text{od} \\
\text{tmp} & \leftarrow \text{sum} \\
\text{for } i & \leftarrow 1 \text{ to } N-1 \text{ do} \\
& \quad \text{MPI\_send(tmp, 1, MPI\_INT, lower neighbour, ...)} \\
& \quad \text{MPI\_recv(tmp, 1, MPI\_INT, higher neighbour, ...)} \\
& \quad \text{sum} \leftarrow \text{sum} + \text{tmp} \\
& \text{od}
\end{align*}
\]

a) Describe how this algorithm works for a \( 3 \times 3 \) torus, where the nodes are labelled rowise (starting with rank 0) from top left to bottom right.

b) Dependent on \( N \) only, determine the number of necessary computing steps (i.e. \( T(p) \)) of the parallel program (communication is to be neglected) and, thus, give an approximation for the speed-up and parallel efficiency. Sketch the parallel efficiency in a small diagram and discuss these results!

c) Implement the above program using MPI and test it for different values of \( N \), running your code on the cluster of the Chair for Computation in Engineering.