Exercise 2

2.1 MPI Ring Communication

The program in Listing 1 was used for measureing the delay between message send and receive. The slurm-batch script in Listing 2 was used to iterate over a different number of processes and messages.

```
int main(int argc,  const char** argv) {

    MPI_Comm_size(MPI_COMM_WORLD, &comm_size);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    // Compute
    uint8_t data = 0;
    MPI_Status status;
    int nummsg = ap.retrieve<int>("nummsg");
    auto starttime = std::chrono::high_resolution_clock::now();
    for (int i = 0; i < nummsg; i++) {
        MPI_Send(&data, 0, MPI_BYTE, (rank+1)%comm_size, 0, MPI_COMM_WORLD);
        MPI_Recv(&data, 1, MPI_BYTE, (rank-1)%comm_size, 0, MPI_COMM_WORLD, &status);
    }
    auto endtime = std::chrono::high_resolution_clock::now();

    MPI::Finalize();

    return 0;
}
```

Listing 1: C++ MPI Implementation for SIMD Ring Communication

```
#! /bin/bash

#SBATCH --ntasks-per-node=6 --ntasks=24
#SBATCH -o out/%A_%a
#SBATCH --array 2-24:2


for nmsg in 100000, 1000000, 10000000
do
    module load mpi
    mpirun -host creek01,creek06,creek05,creek04 -np ${SLURM_ARRAY_TASK_ID} ./ring.out -n $nmsg
done
```

Listing 2: SBATCH file for job tests

These iterations rsulted in the observed per message delays as visulized in Figure 1.
The optimal mapping with 12 processes looks ideally like in Figure 2.

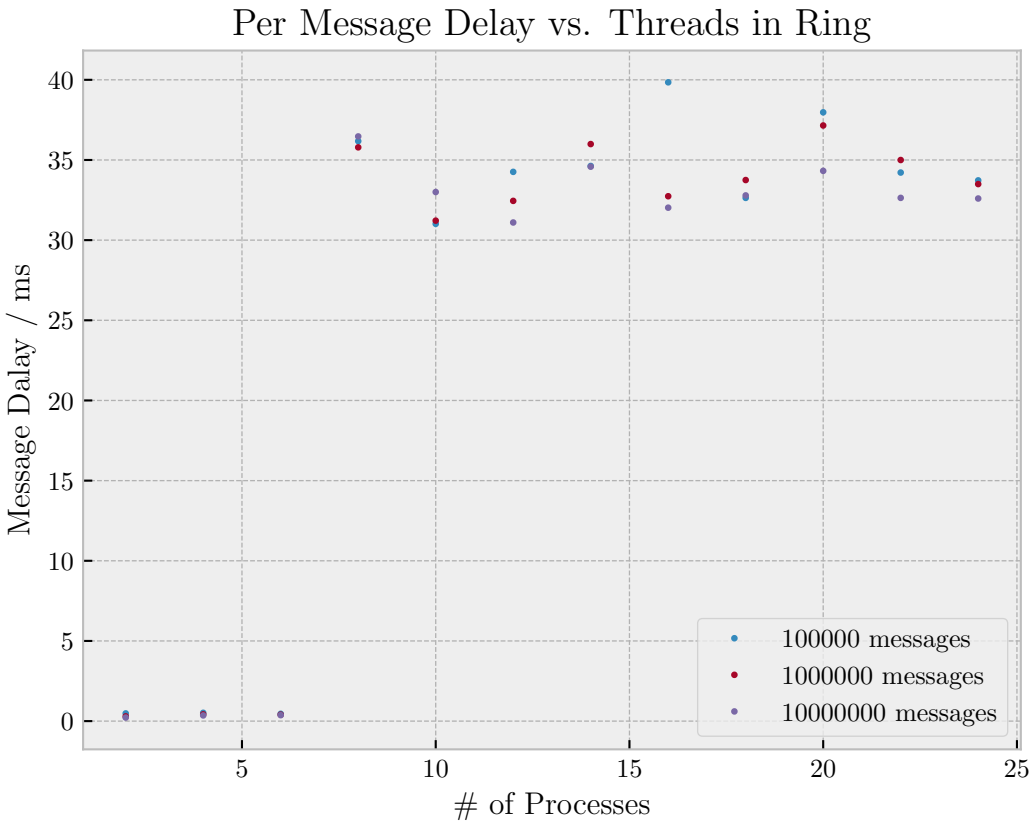

Figure 1: img/Ring

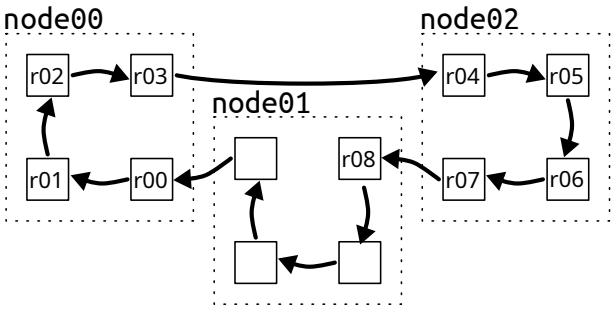

Figure 2

Here each node has 4 processes (one per physical CPU core) running simultaneously, requiring three nodes. When communicating there are two different connections:

- in-node
- and node-to-node

While the in-node connections require a relatively small latency for message passing ($\approx 1\,\mathrm{ms}$), the messages passed between nodes require longer to send.

To optimize the general message delay the number of node-to-node connections have to be minimized, which is three in this case. Therefore only three in 12 messages passed per cycle require longer transmission times.

Assuming these times are constant, then we can assume that the following equeation can approximate the Messagedelay between nodes

$$t_{\text{in-node}} \approx 1\,\text{ms} \tag{1}$$

$$t_{\text{avg, 2 n2n conn}} \approx 36\,\text{ms} \tag{2}$$

$$= \frac{1}{4}t_{\text{n2n}} + \frac{3}{4}t_{\text{in-node}} \tag{3}$$

$$\Rightarrow t_{\text{n2n}} \approx 141\,\text{ms} \tag{4}$$

⇒ This calculation shows that node-to-node communication is two magnitudes slower then inter-node communication.