

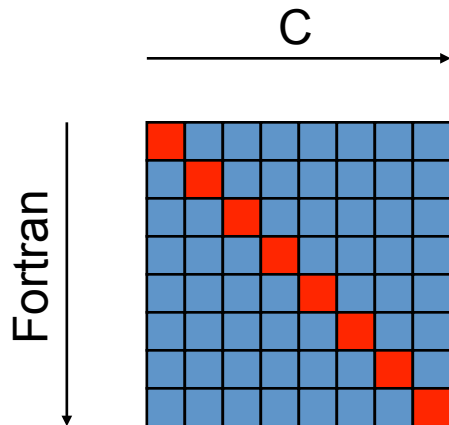
# Homework 3

Message Passing hw #1

Deadline: Feb 25 2015

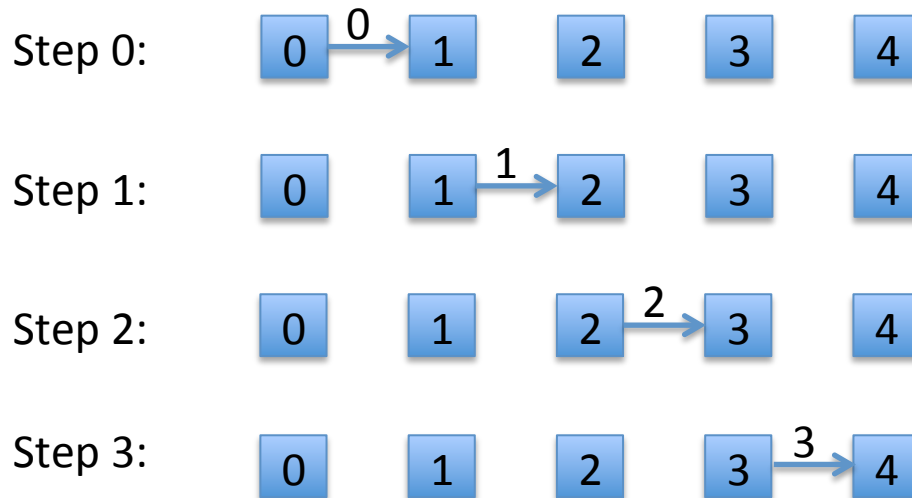
# MPI Datatype

- Create a datatype describing a matrix diagonal
- What's different between the C and the Fortran version of the datatype?
- Using only MPI features provide a simple solution for a matrix transpose.



# Ring Application

- Write a simple application that passes a token between all the processes in one communicator.



# Process Grids

- Imagine a matrix distributed over a 2D block-cyclic block-cyclic distribution
- Use the minimum number of MPI functions to create row and column communicators

0	1	2	0	1	2
3	4	5	3	4	5
6	7	8	6	7	8

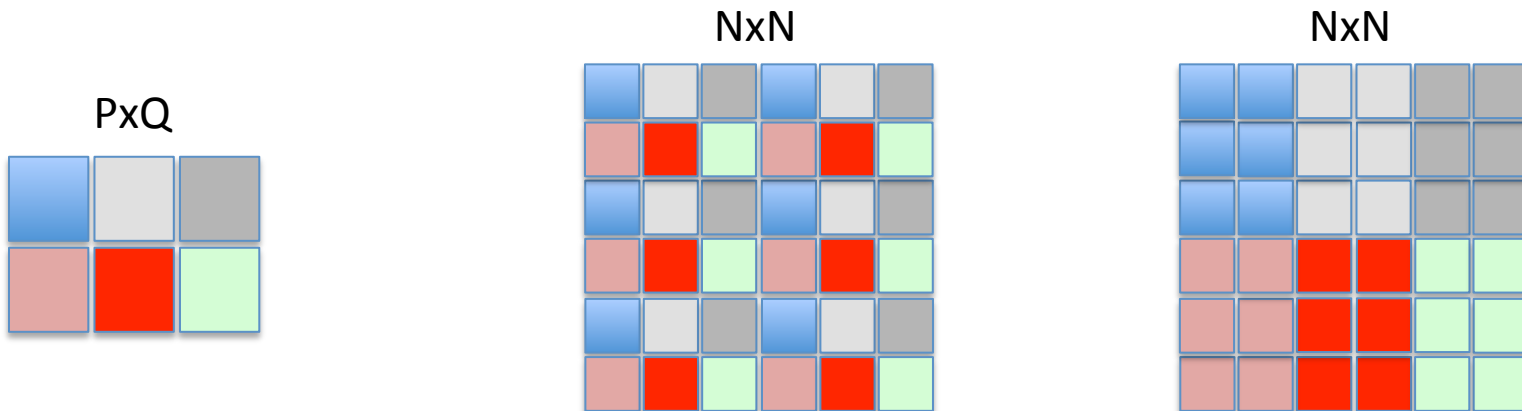
0	1	2	0	1	2
3	4	5	3	4	5
6	7	8	6	7	8

# Multi-pipeline communications

- Merge the first and second part of this homework to create an example of application where everybody has a token originally and they pass it around in each dimension.
  - The token received from the west is passed to the east at the next iteration, while the token received from the north is passed to the south at the next iteration
  - How many steps we need to have a full exchange (e.g. every process have seen all tokens on it's line and column of processors).

# PDGEMM

- Use the previous application to implement a parallel version of the matrix matrix multiplication using MPI (use well-known algorithms: SUMMA or PUMMA).
- Suppose the  $N \times N$  matrices are distributed in cyclic( $k$ )/cyclic( $k$ ) way on the  $P \times Q$  processor grid.  $k$  is a user supplied argument. All matrices have the same distribution.
- On each node the data is stored in memory in LAPACK format.



# What to turn in

- Source code and a Makefile
  - Make sure the code works as expected (check it with bi/tri-diagonal matrices as an example)
- For the local computation you can use the BLAS flavor available to you (MKL, ATLAS ...)
- A pdf describing your findings. Extra credits for:
  - Show the impact of the blocking factor on the performance
  - Try different communication patterns
  - Compare with PBLAS version of GEMM