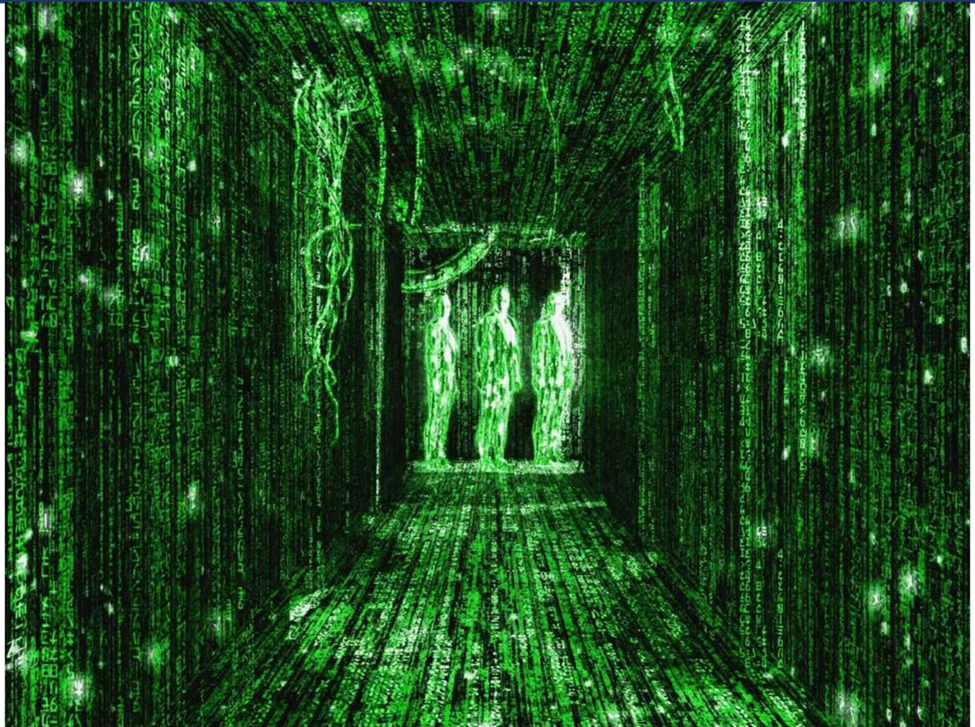
Parallel programming

Matrix Algorithms
in
OpenMP and MPI



Today's topic

- Coding seminar
- Goals:
  - Practice the theory from the lectures
  - Practice OpenMP and MPI
- 4 Tasks
  - 2x OpenMP
  - 2x MPI

Task dependencies

- Calculate matrix G as:
  - $G = \left((A * B) * (C * A)\right) + (D * E) + F$
- Create task dependency graph and write parallel version of the code respecting task dependencies
- Use Instruction Level Parallelism
- Use OpenMP and the provided template

LU Factorization

- Compute LU Factorization of matrix A:
  - $A = L * U$
  - Where L is lower triangular matrix and U is upper triangular matrix
- Use following rules:
  - $L_{00}.U_{00} = A_{00}$
  - $L_{10}.U_{00} = A_{10}$
  - $L_{00}.U_{01} = A_{01}$
  - $L_{10}.U_{01} + L_{11}.U_{11} = A_{11}$

$$\begin{array}{|c|c|}\hline A_{00} & A_{01} \\ \hline A_{10} & A_{11} \\ \hline \end{array} = \begin{array}{|c|c|}\hline L_{00} & 0 \\ \hline L_{10} & L_{11} \\ \hline \end{array} * \begin{array}{|c|c|}\hline U_{00} & U_{01} \\ \hline 0 & U_{11} \\ \hline \end{array}$$

- Use OpenMP and write a parallel code
  - Experiment with scheduling policies
  - Experiment with chunk sizes
  - Experiment with different methods of parallelization

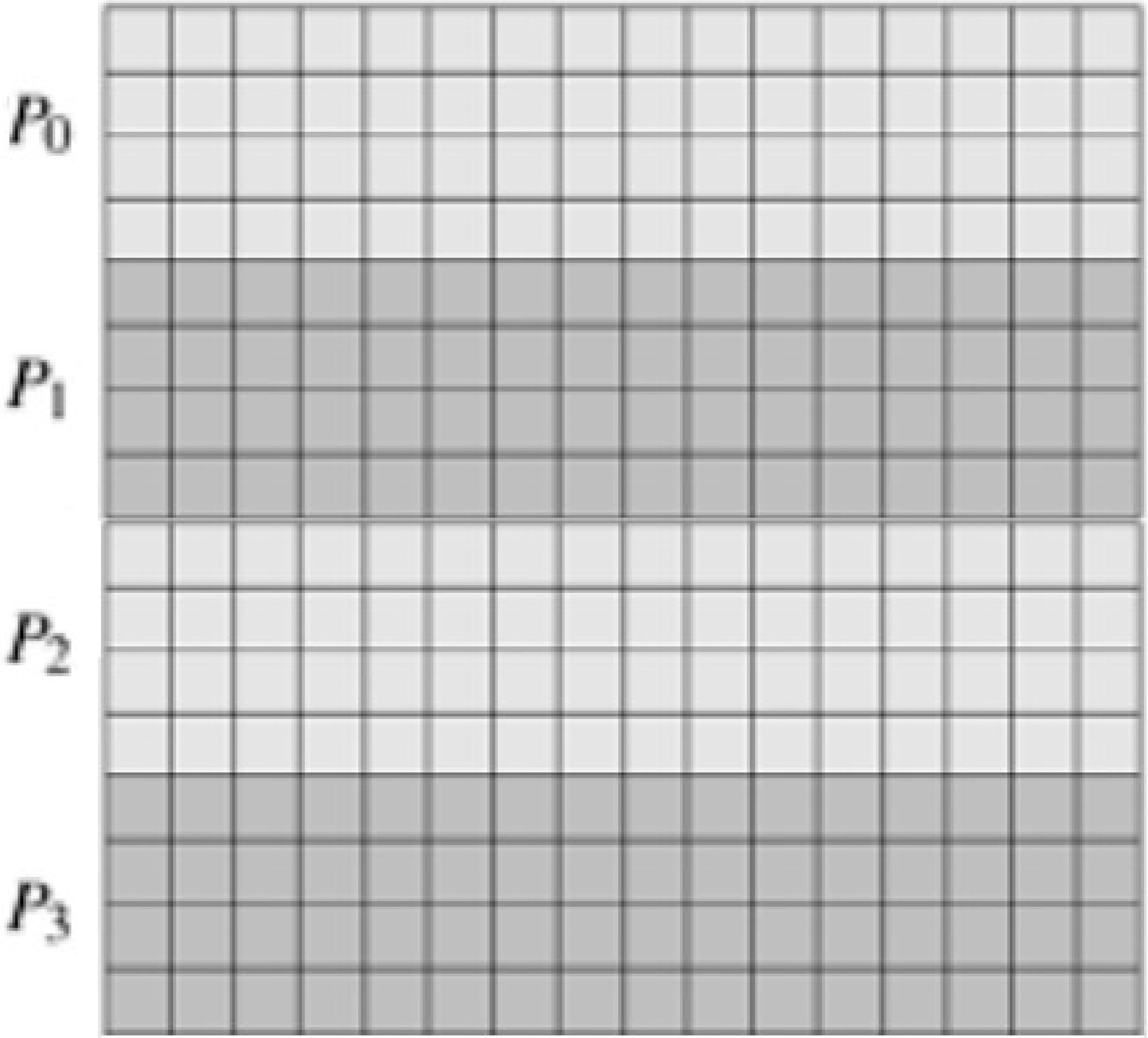
Example of LU decomposition

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Gauss elimination

- Calculate Gauss Elimination of matrix A.
- Write parallel version of the code using MPI.
  - Split Matrix rows into P **distinct** blocks where P is the number of processes.
- Measure the speedup for different number of processors.
- Use row-wise distribution

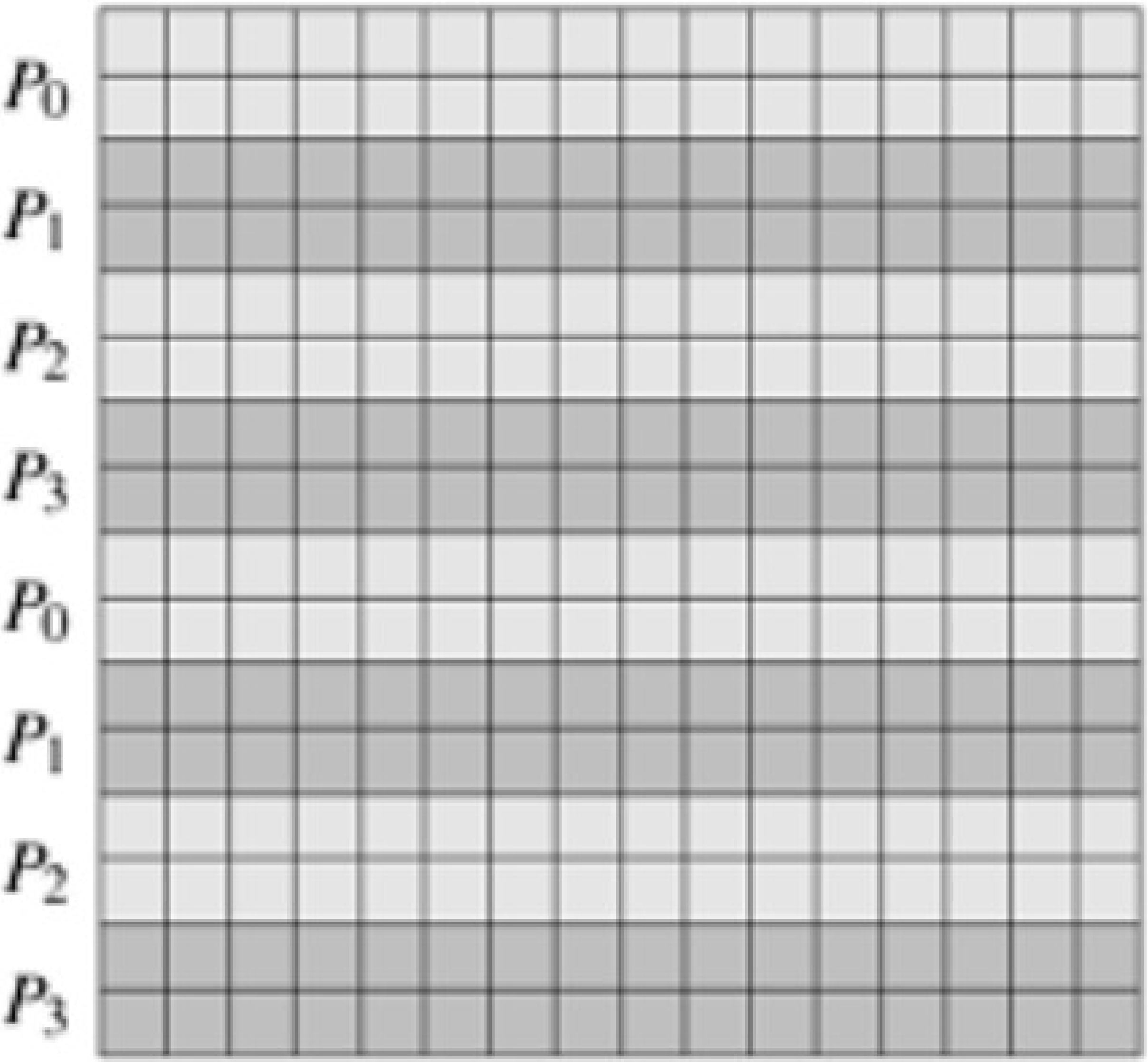
Row-wise distribution



Cyclic Gauss elimination

- Calculate Cyclic Gauss Elimination of matrix A.
- Write parallel version of the code using MPI.
  - Split Matrix rows into P **intersected** blocks where P is the number of processes.
- Measure the speedup for different number of processors.
- Use cyclic block distribution

Cyclic Block Distribution


P_0
P_1
P_2
P_3
P_0
P_1
P_2
P_3