An Educational Module Illustrating how Sparse Matrix-Vector Multiplication on Parallel Processors Connects to Graph Partitioning

H. M. Bücker, M. A. Rostami
Institute for Computer Science
Friedrich Schiller University Jena

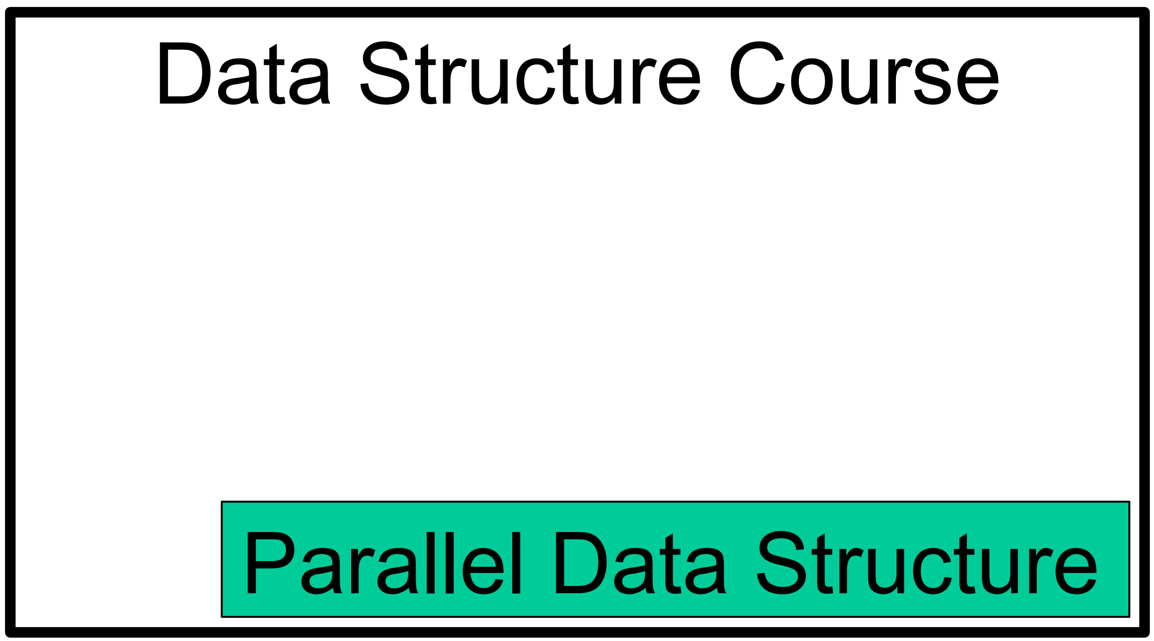
Friedrich-Schiller-Universität Jena

Outline

- Motivation
- Parallel Sparse Matrix-Vector Product
- Graph Partitioning
- **EXPL**oring **A**lgorithms **IN**teractively (EXPLAIN)
- Concluding Remarks

Motivation

- Integrating parallelism into an existing curriculum is easier than modifying curriculum or setting up new curriculum
- Not completely new course



Sparse Matrix-Vector Product

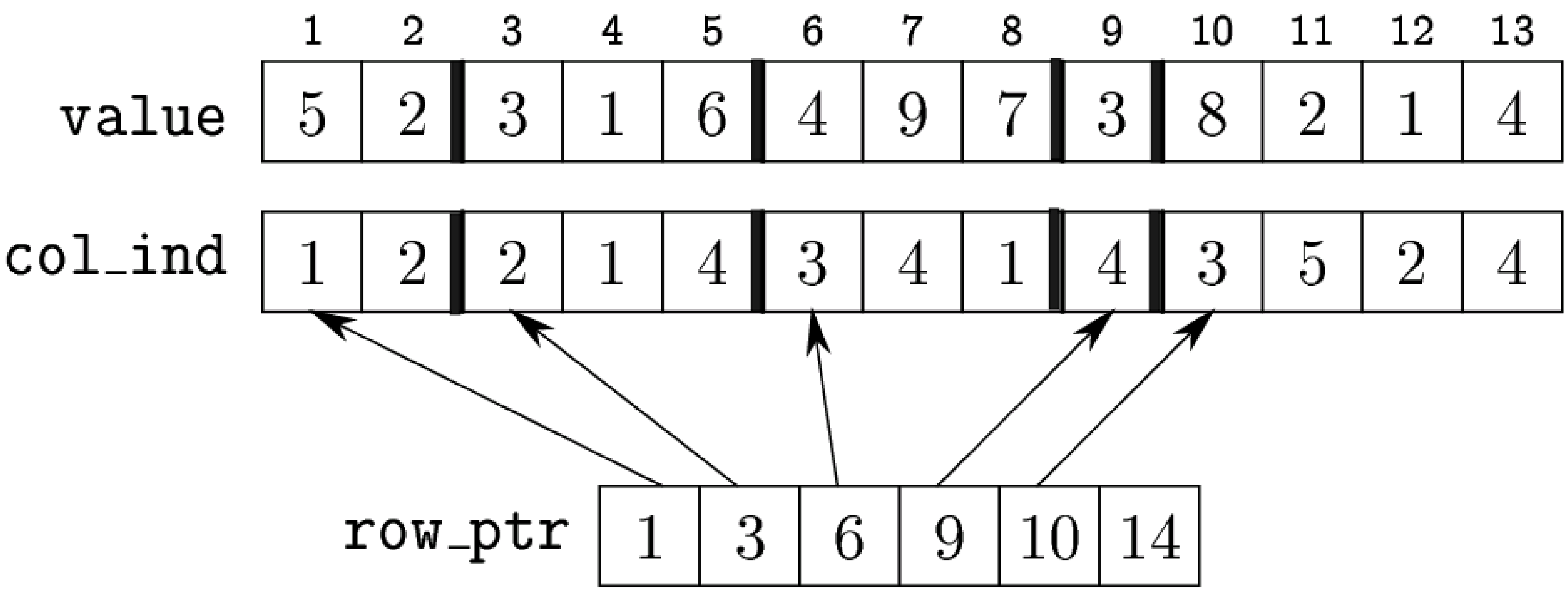
$A \in R^{n \times n}$ Large, Sparse,
Known Symmetric Pattern

$x \in R^{n}$ Dense

Find $y = Ax$

Data Structure (CRS)

$$A = \begin{bmatrix} 5 & 2 & 0 & 0 & 0 \\ 1 & 3 & 0 & 6 & 0 \\ 7 & 0 & 4 & 9 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 1 & 8 & 4 & 2 \end{bmatrix}$$



Complexity of Serial Algorithms

- n^2 if A dense
- n if A sparse and each row of A has at most constant number of nonzeros

Parallel Sparse Matrix-Vector Product

Given $A \in R^{n \times n}$, $x \in R^n$

+ Having p processors

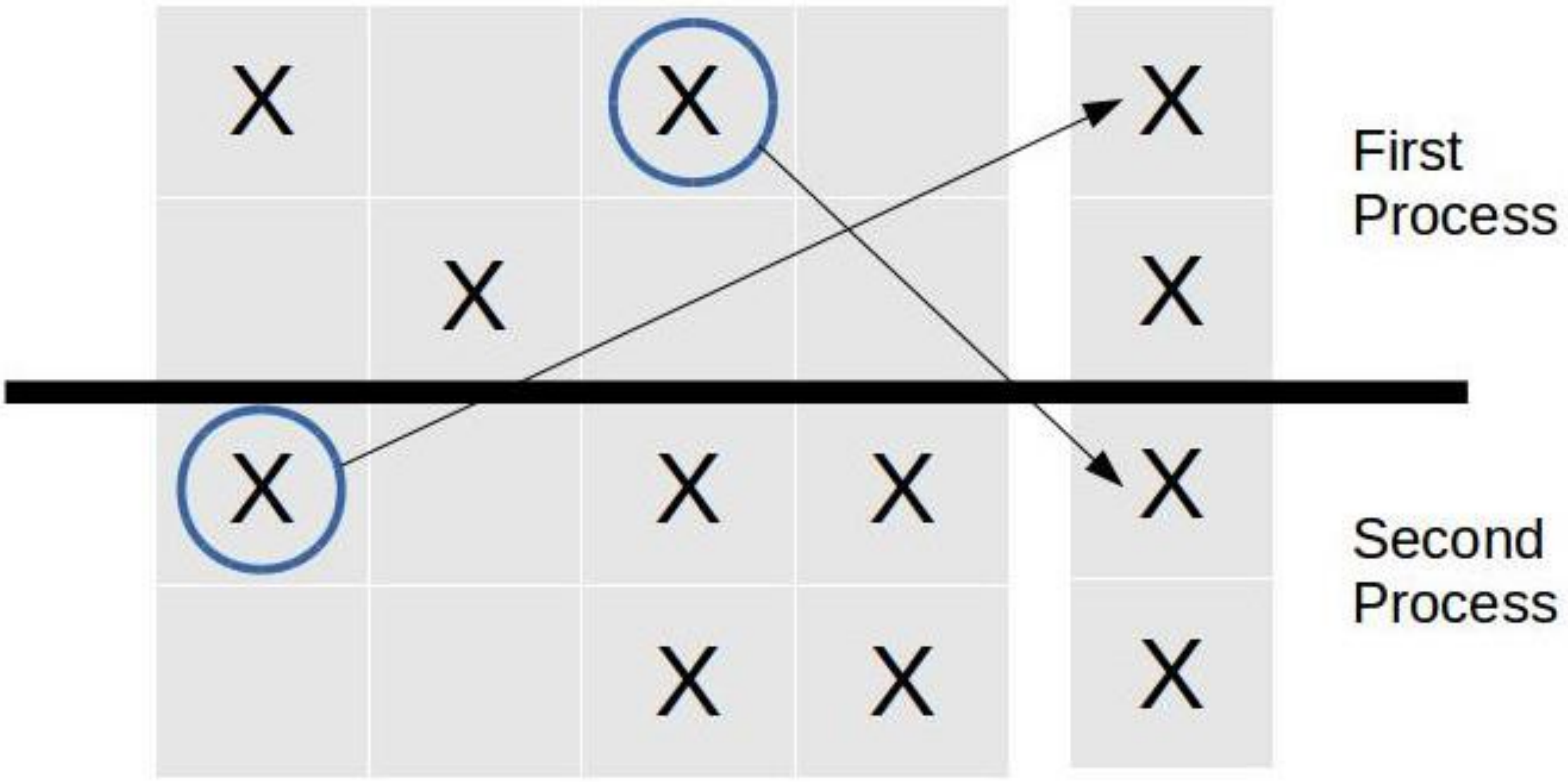
Compute $y = Ax$

Parallel Data Distribution

Assumptions:

- Nonzeros of a complete row are not distributed to more than one process
- Consistent: if nonzeros of row i are on process p, then x_i and y_i are also on p.

Communication Needed!



Problem 1: Optimal Data Distribution

- How do we distribute data to processes such that communication is minimized and operations on each process are (almost) balanced?
- NP-complete Problem

Graph Interpretation

- An undirected graph $G = (V, E)$
- $V = \{1,2, \dots, n\}$ for each row/col of A
- $E = \{(i, j) | i, j \in V \text{ and } a_{\{ij\}} \neq 0 \text{ for } i > j\}$
- Partition $P: V \rightarrow \{1,2, \dots, p\}$ represents distribution of vertices to p processors

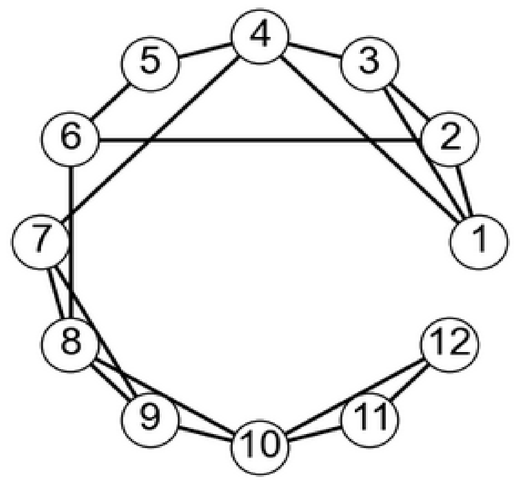
Problem 2: Graph Partitioning

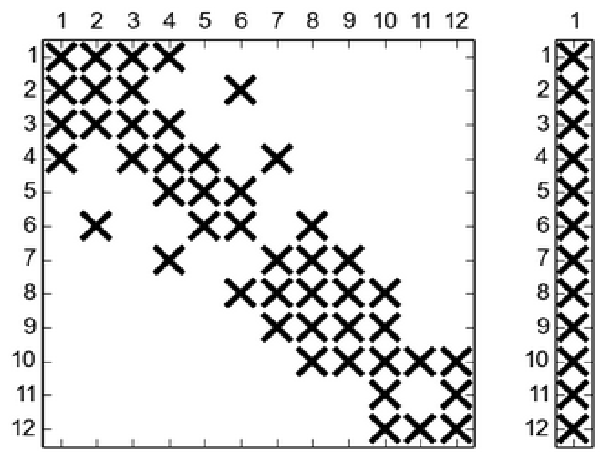
- Find partition $P: V \to \{1,2,\ldots,p\}$
  such that
  $\mathrm{cut}(P)$ is minimized and
  $|V_1| \cong |V_2| \cong \cdots \cong |V_p|$.

- Problem 1 (Linear Algebra) is equivalent to Problem 2 (Graph Theory).

EXPLAIN

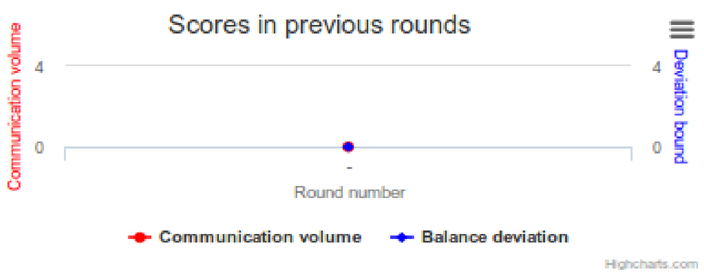
EXPLAIN Parallel Matrix-Vector Product - Round 1



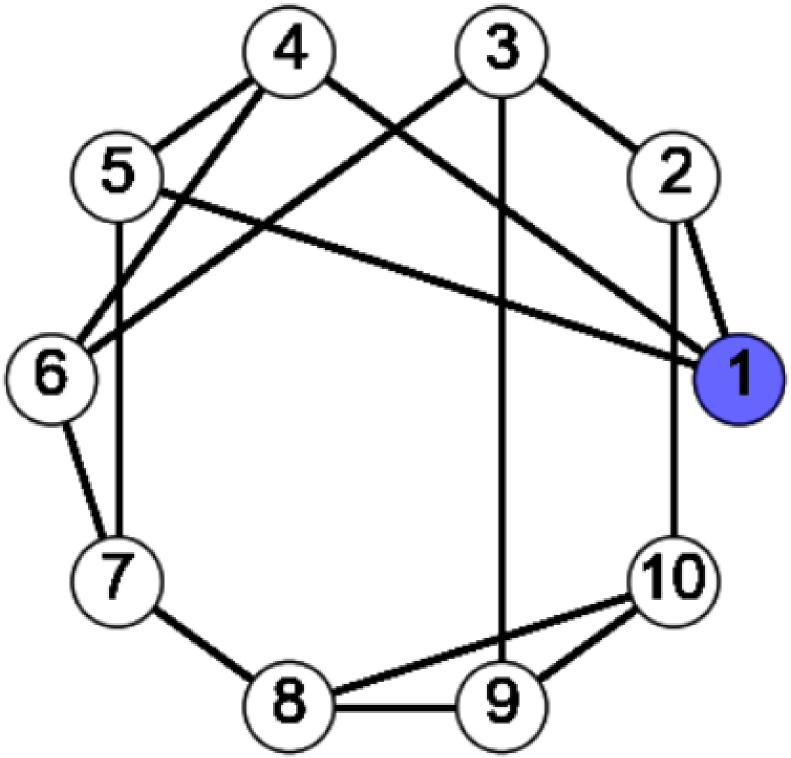


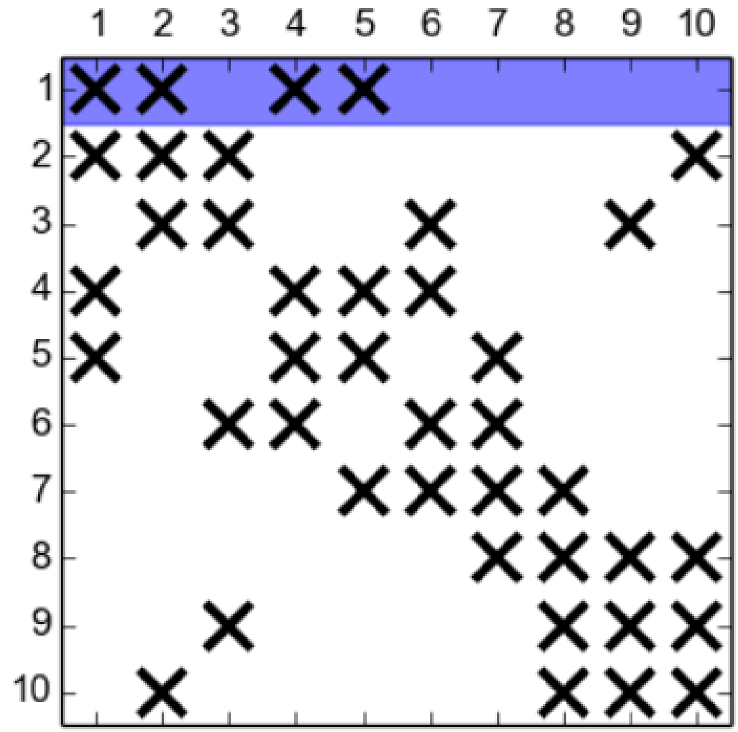
Color of processes:

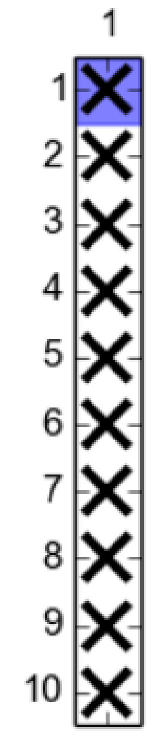
Order of selection:



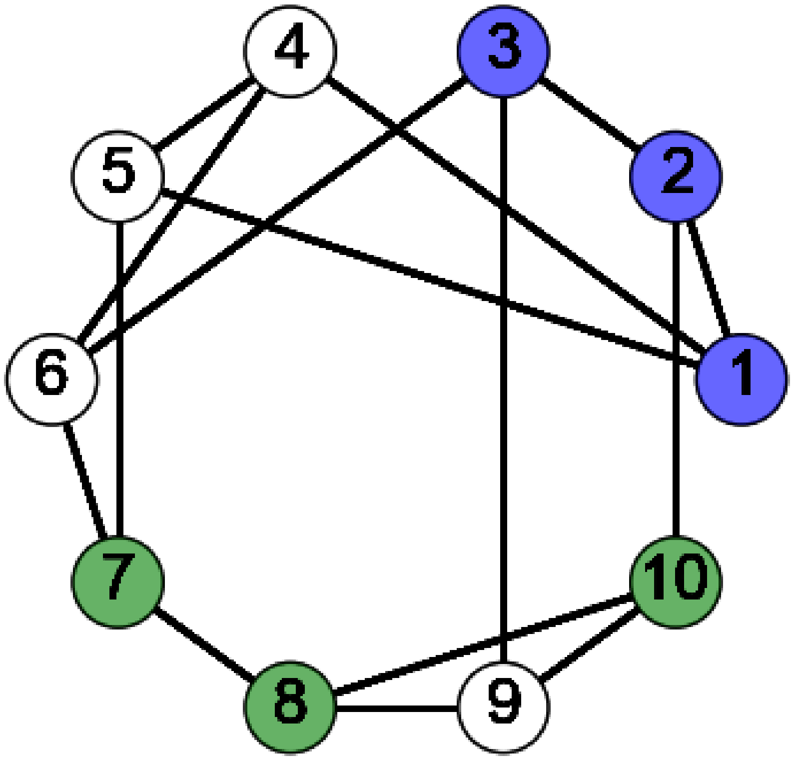
Assign Rows to a Process

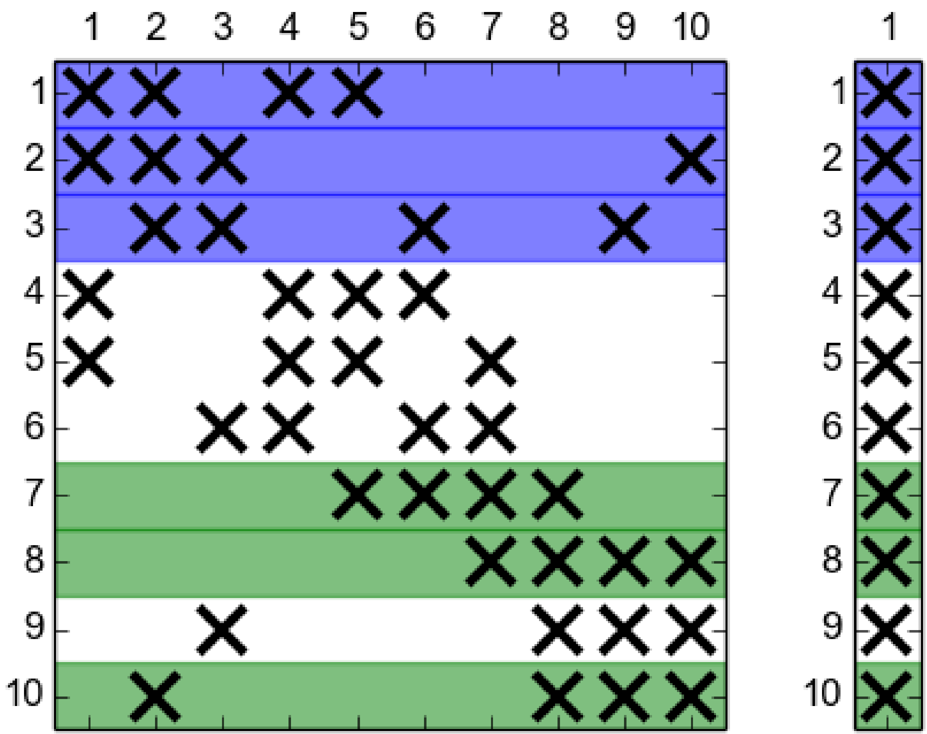




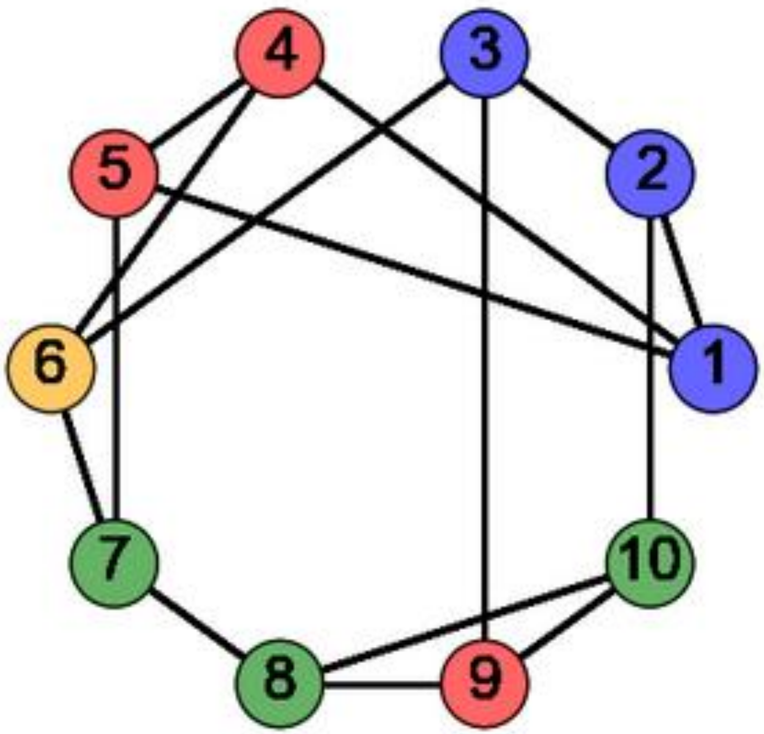


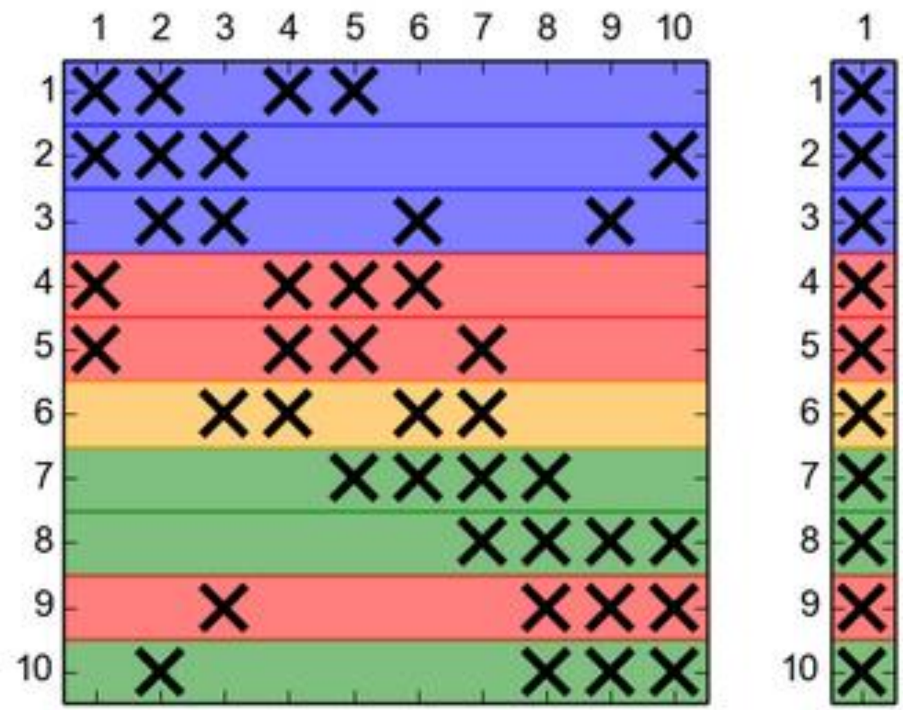
Assigning Rows to Two Processes



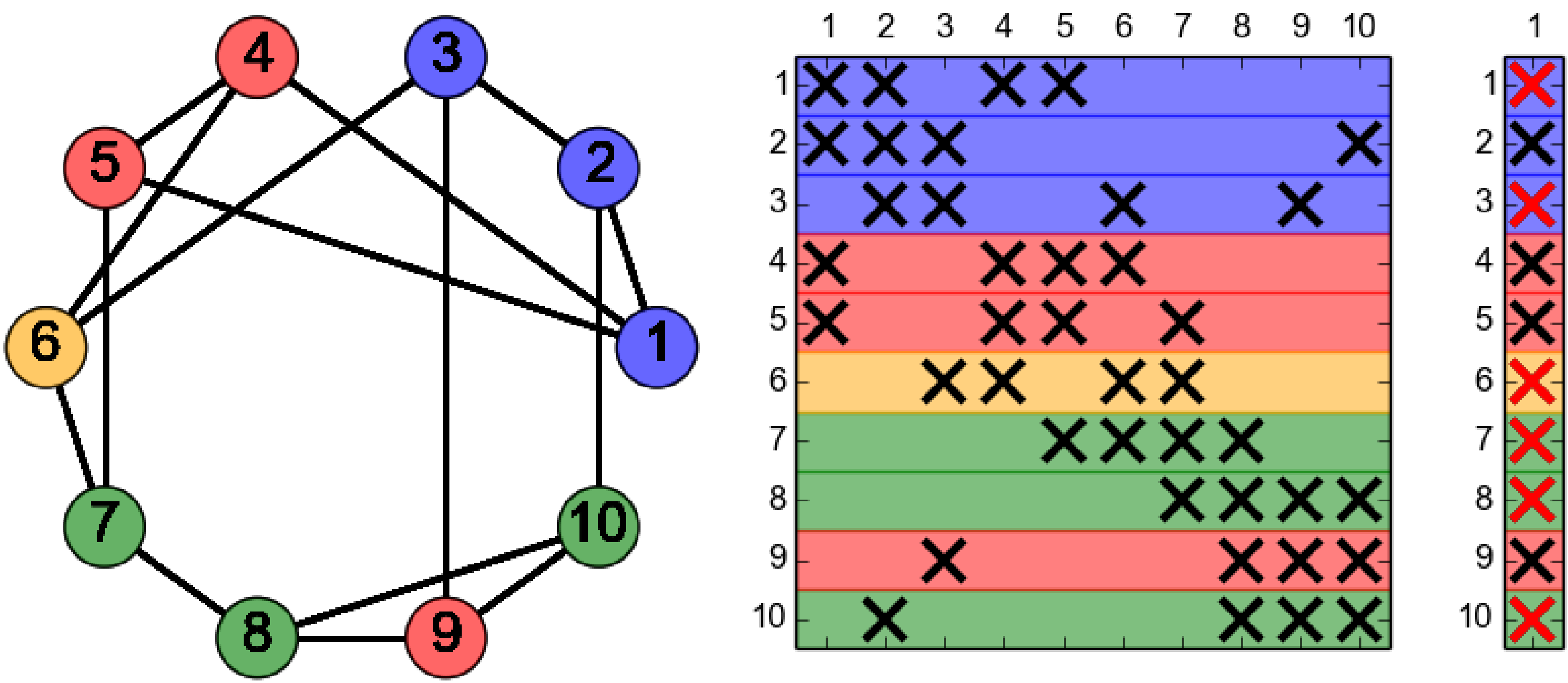


Clear View of Communication





All Communications for Red Processor



Concluding Remarks

- Matrix-vector product is simple problem
- Sparsity offers rich set of (serial) data structures
- Parallel sparse matric-vector product is pedagogically interesting because transition from serial to parallel leads to additional aspects (minimizing and balancing) and also increases complexity

Concluding Remarks

- Straightforward integration of parallelism into an existing course
- EXPLAIN:
  - Linear algebra vs graph theory
  - Clear visualization of problem
  - High level of interactivity