Parallel and Distributed Computing

CS3006

Lecture 7

Mapping Schemes

21st February 2024

Dr. Rana Asif Rehman

Agenda

- A Quick Review
  - **Decomposition Techniques**
- **Distribution Schemes**
  - Block-distribution
    - Row-wise
    - Column-wise
    - 1D and 2D
  - Cyclic and Block-cyclic

Mapping Techniques

- Once a problem has been decomposed into concurrent tasks, these must be mapped to the processes (that can be executed on a parallel platform)
- Mapping must minimize the overheads
  - Communication
  - Idling
- Assigning all work to one processor trivially minimizes communication at the expense of significant idling

Mapping Schemes

Static Mapping

- Distributing the tasks among the processes before execution of the program
  - E.g., usually used in situation where total number of tasks and their sizes are known before the execution of the program
- Easy to implement in massage passing paradigm

Dynamic Mapping

- When total number of tasks are not known a priori
- (OR) when task sizes are unknown
  - In this case static mapping can lead to serious load-imbalances.

Both static and Dynamic Mappings are equally easy in shared memory paradigm

Schemes for Static Task-Process Mapping

(Mappings based on Data Partitioning)

Array Distribution Schemes

- In a decomposition based on partitioning data, **mapping** the relevant **data** onto the processes is equivalent to **mapping tasks** onto processes *
- Commonly used array mapping schemes:
  - Block distribution
    - 1D and 2D
  - Cyclic and block-cyclic distribution
  - Randomized block distribution

Schemes for Static Mapping

(Mappings based on Data Partitioning)

Array Distribution Schemes:

- Block distribution (1D)

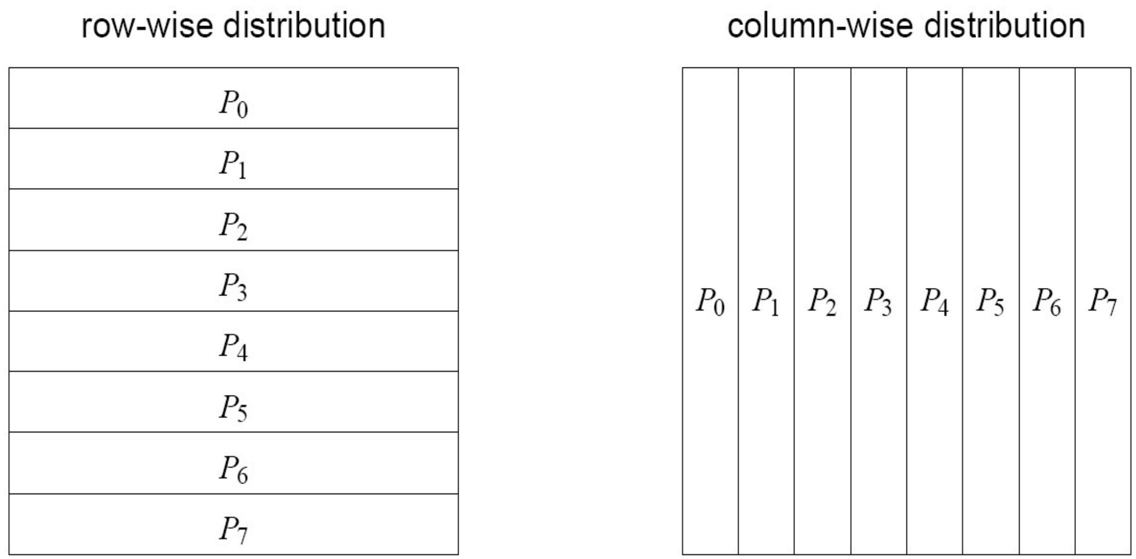


Figure 3.24 Examples of one-dimensional partitioning of an array among eight processes.

Schemes for Static Mapping
(Mappings based on Data Partitioning)

Array Distribution Schemes:

- Block distribution (2D)

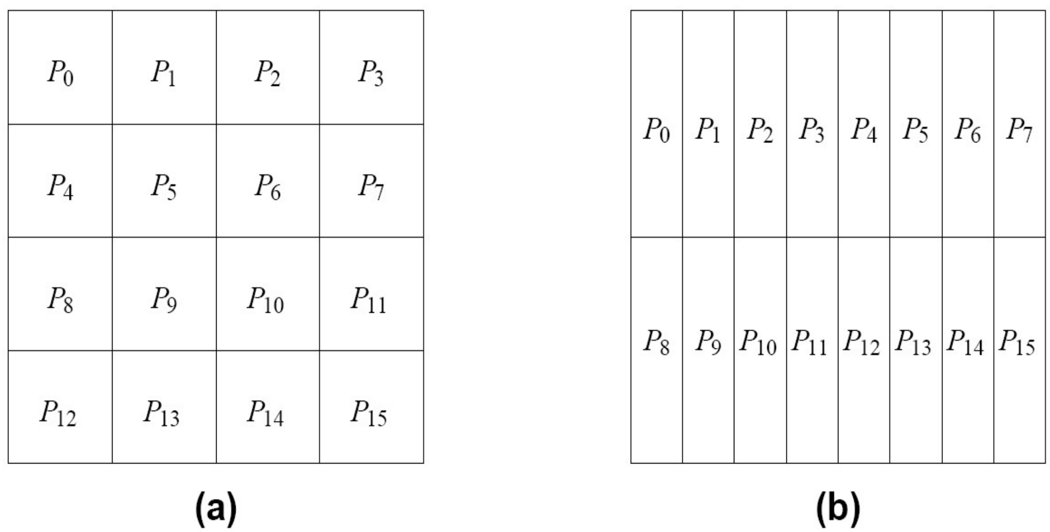


Figure 3.25 Examples of two-dimensional distributions of an array, (a) on a 4×4 process grid, and (b) on a 2×8 process grid.

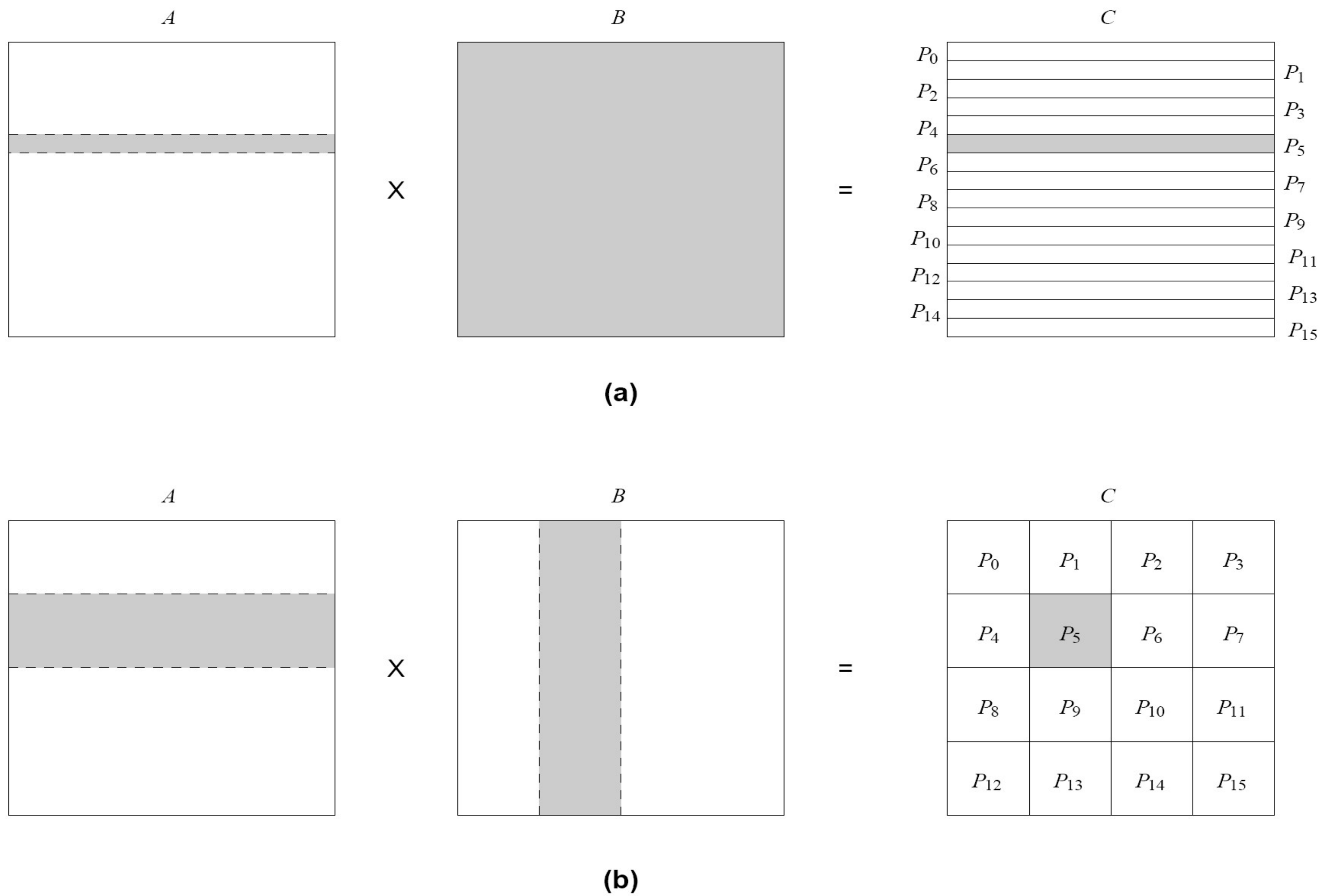


Figure 3.26 Data sharing needed for matrix multiplication with (a) one-dimensional and (b) two-dimensional partitioning of the output matrix. Shaded portions of the input matrices A and B are required by the process that computes the shaded portion of the output matrix C.

Schemes for Static Mapping

(Mappings based on Data Partitioning)

Array Distribution Schemes:

- Cyclic distribution (HERE array size=4 x 4 and p=3)

P0	**P1**	**P2**	**P0**
P1	**P2**	**P0**	**P1**
P2	**P0**	**P1**	**P2**
P0	**P1**	**P2**	**P0**

Schemes for Static Mapping
(Mappings based on Data Partitioning)

Array Distribution Schemes:

- Block-Cyclic distribution (1D and 2D)

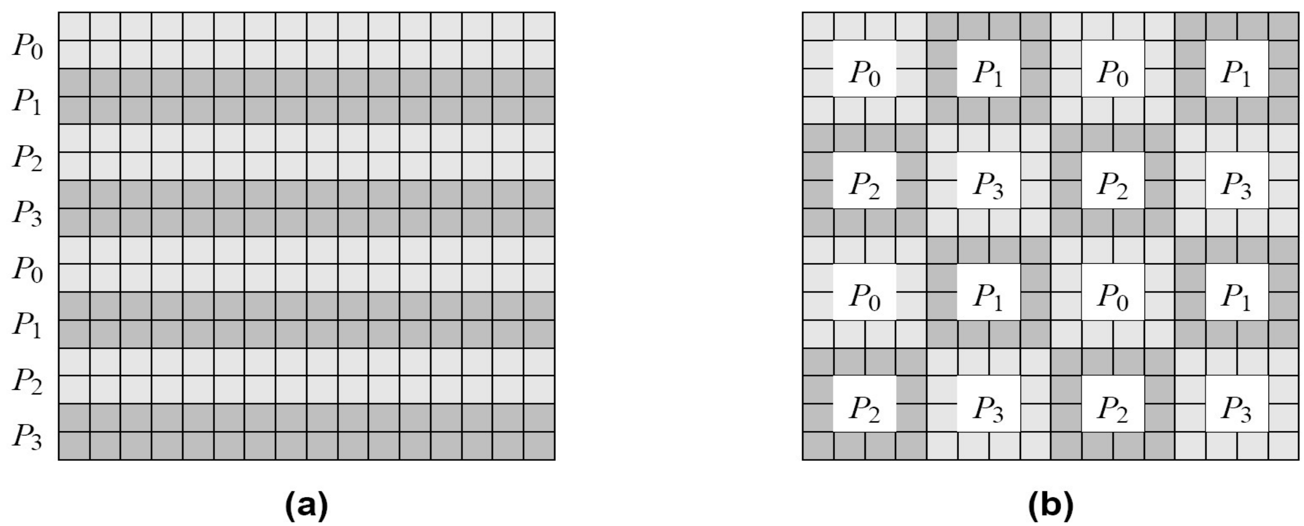


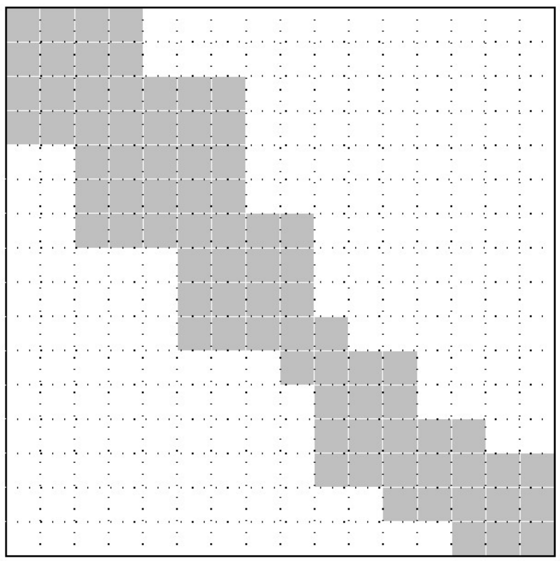
Figure 3.30 Examples of one- and two-dimensional block-cyclic distributions among four processes. (a) The rows of the array are grouped into blocks each consisting of two rows, resulting in eight blocks of rows. These blocks are distributed to four processes in a wraparound fashion. (b) The matrix is blocked into 16 blocks each of size 4×4, and it is mapped onto a 2×2 grid of processes in a wraparound fashion.

Schemes for Static Mapping

(Mappings based on Data Partitioning)

Array Distribution Schemes:

- Block-Cyclic distribution (Issue)



(a)

P_0	P_1	P_2	P_3	P_0	P_1	P_2	P_3
P_4	P_5	P_6	P_7	P_4	P_5	P_6	P_7
P_8	P_9	P_{10}	P_{11}	P_8	P_9	P_{10}	P_{11}
P_{12}	P_{13}	P_{14}	P_{15}	P_{12}	P_{13}	P_{14}	P_{15}
P_0	P_1	P_2	P_3	P_0	P_1	P_2	P_3
P_4	P_5	P_6	P_7	P_4	P_5	P_6	P_7
P_8	P_9	P_{10}	P_{11}	P_8	P_9	P_{10}	P_{11}
P_{12}	P_{13}	P_{14}	P_{15}	P_{12}	P_{13}	P_{14}	P_{15}

(b)

Figure 3.31 Using the block-cyclic distribution shown in (b) to distribute the computations performed in array (a) will lead to load imbalances.

Schemes for Static Mapping

(Mappings based on Data Partitioning)

Array Distribution Schemes:

- Randomized-Block distribution (solution: 1D)

$$V = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]$$

$$\text{random}(V) = [8, 2, 6, 0, 3, 7, 11, 1, 9, 5, 4, 10]$$

$$\text{mapping} = \underbrace{8\ \ 2\ \ 6}_{P_0}\ \ \underbrace{0\ \ 3\ \ 7}_{P_1}\ \ \underbrace{11\ 1\ \ 9}_{P_2}\ \ \underbrace{5\ \ 4\ 10}_{P_3}$$

Figure 3.32 A one-dimensional randomized block mapping of 12 blocks onto four process (i.e., $\alpha = 3$).

Schemes for Static Mapping

(Mappings based on Data Partitioning)

Array Distribution Schemes:

- Randomized-Block distribution (solution: 2D)

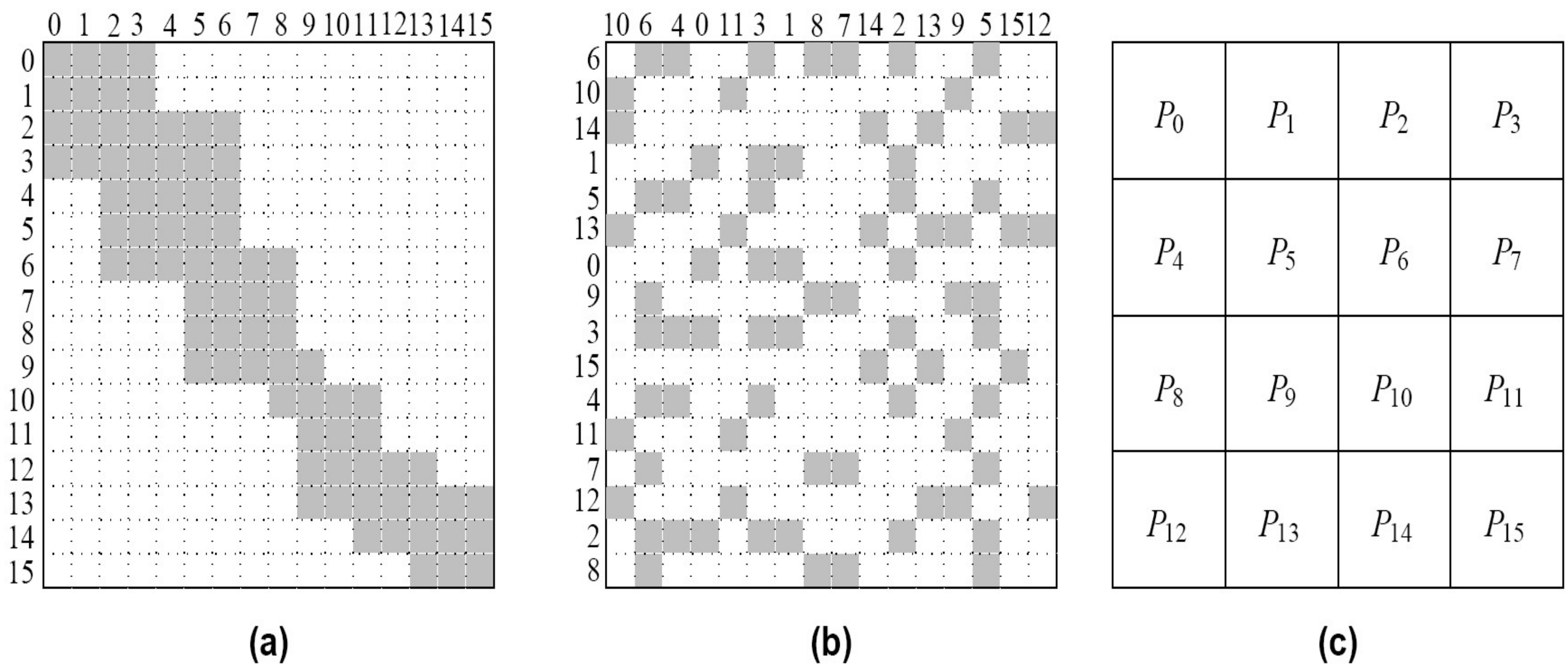


Figure 3.33 Using a two-dimensional random block distribution shown in (b) to distribute the computations performed in array (a), as shown in (c).

Why randomized bock cyclic distribution is not always used?

A simulation model (using a mesh of tasks) for finding dispersion of water contaminant in a lake at different intervals of time.

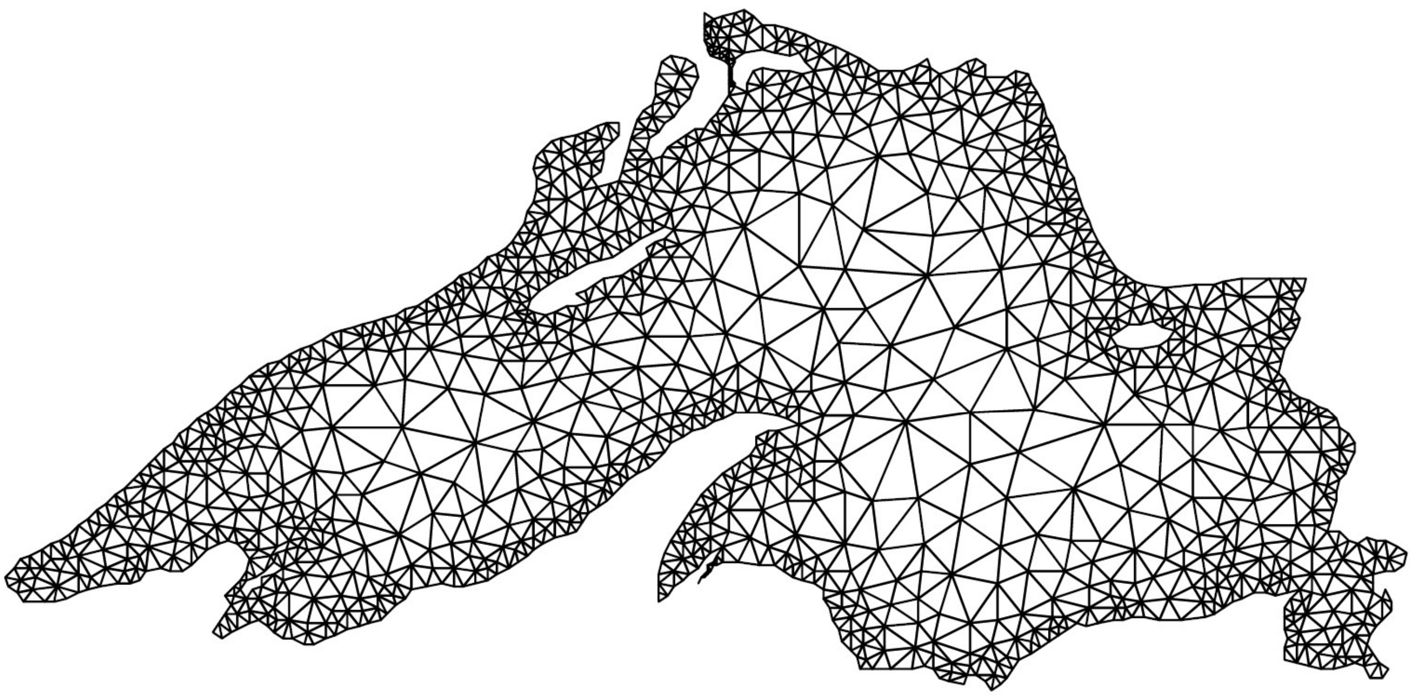


Figure 3.34 A mesh used to model Lake Superior.

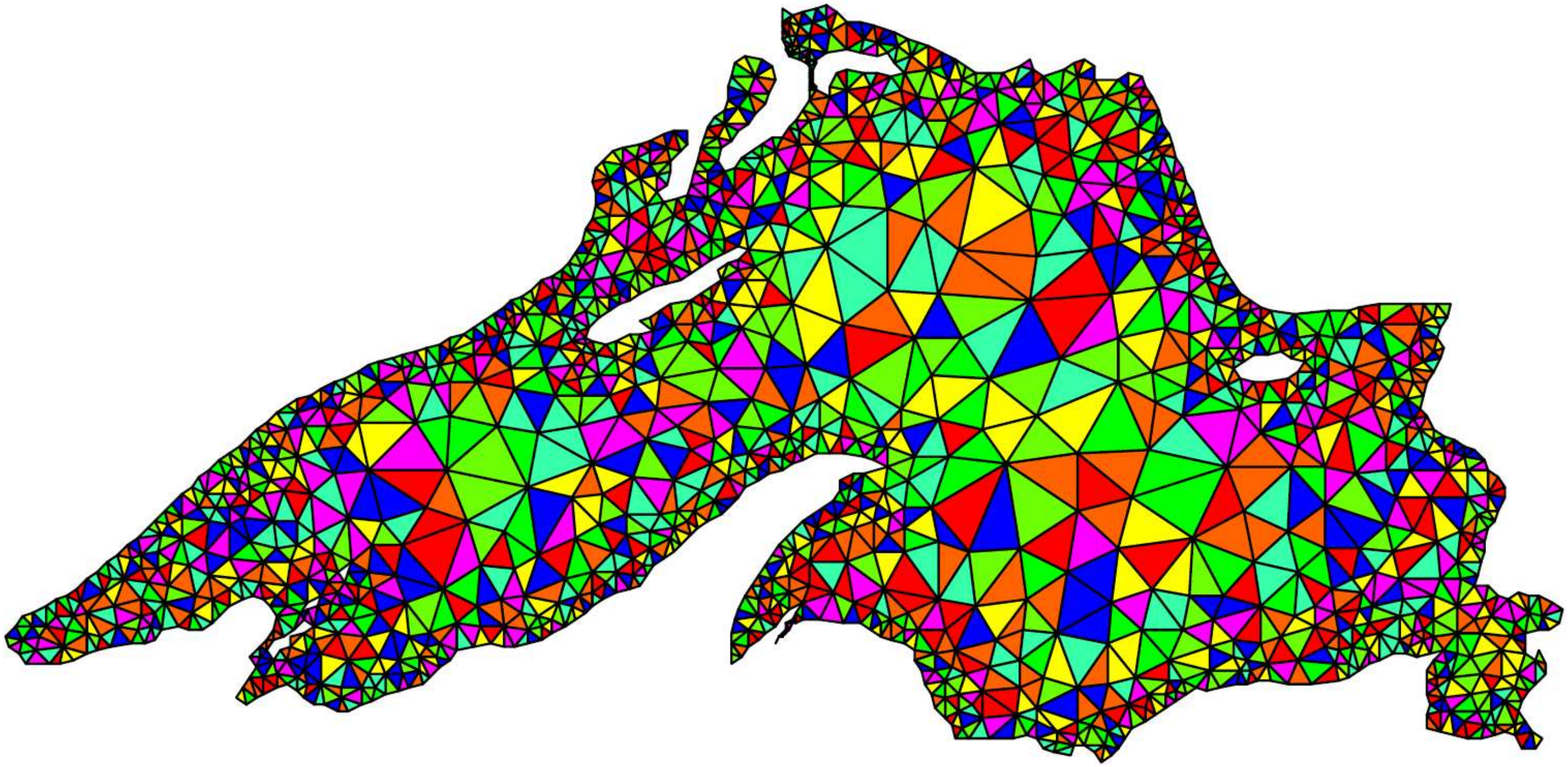


Figure 3.35 A random distribution of the mesh elements to eight processes.

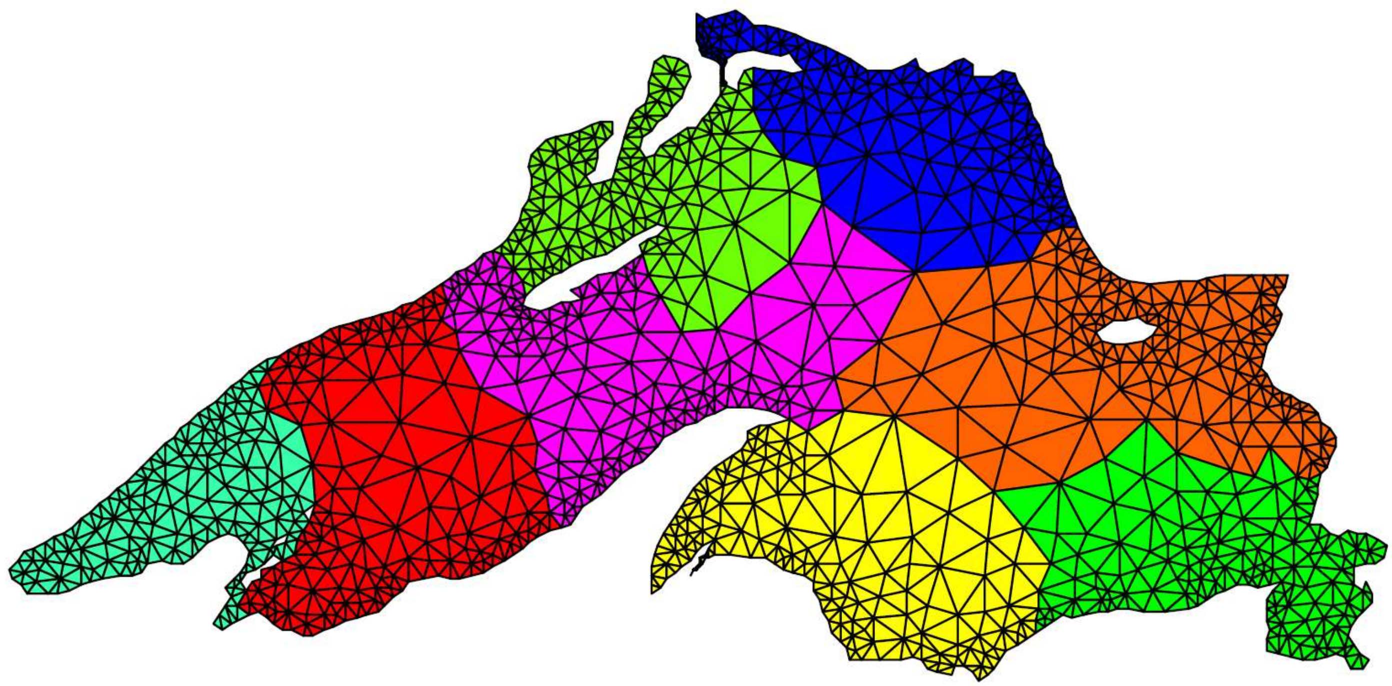


Figure 3.36 A distribution of the mesh elements to eight processes, by using a graph-partitioning algorithm.

Questions

References

1. Kumar, V., Grama, A., Gupta, A., & Karypis, G. (1994). *Introduction to parallel computing* (Vol. 110). Redwood City, CA: Benjamin/Cummings.
2. Quinn, M. J. Parallel Programming in C with MPI and OpenMP,(2003).