PARALLEL PROCESSING

Mohammed Alabdulkareem

kareem@ksu.edu.sa

Office 2247

12

DATA DECOMPOSITION

- Data decomposition is used with problems that operate in large amounts of data, in systems where memory architecture plays a performance role.
- Data decomposition may be based on input data or output data.

DATA DECOMPOSITION

- Steps for decomposing data:
  - 1. Identify the data used and/or produced in the computations (input, output, both).
  - 2. Divide this data across tasks (linear/geometric or recursive decomposition).
  - 3. Obtain a computational partitioning that corresponds to the data partitioning.
  - 4. In distributed memory architectures, add the necessary data allocation and movement actions.

DATA DECOMPOSITION

- General guidelines when decomposing data:
  - Generate comparable amounts of work (for load balancing).
  - Maximize data locality (or minimize the need of task interactions); i.e., minimize volume of data involved in task interactions, minimize frequency of iterations, minimize contention and hot spots.
  - Overlap computation with interactions to hide interaction effect.

DATA DECOMPOSITION

- Minimize interaction overhead (1):
  - Minimize the volume of data exchange considering the cost associated with each word that is communicated.
  - Minimize the frequency of interactions considering the startup cost associated with each interaction (try to merge multiple interactions into one, when possible).
  - Overlap computations with interactions by using non-blocking communications i.e., non-blocking operations (MPI_Isend and MPI_Irecv).

DATA DECOMPOSITION

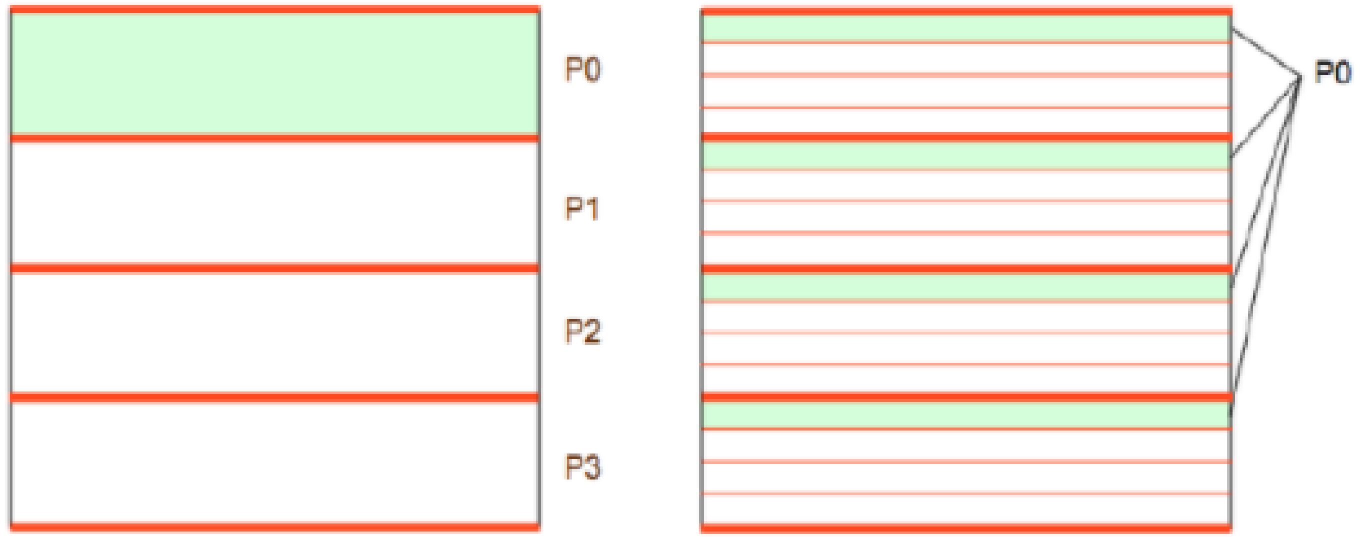
- Minimize interaction overhead(2) :
  - Use collective communications instead of point-to-point primitives (this also provides programming simplicity).
  - Minimize contention and hot-spots by using decentralized techniques or by replicating data whenever it is necessary.
  - Replicate communications

DATA DECOMPOSITION

- Interaction patterns:
  - Point to Point (one to one).
  - Scatter and broadcast (one to all).
  - Gather and Reduce (all to one).
  - All to All (each processor sends its data to all others)

DATA DECOMPOSITION

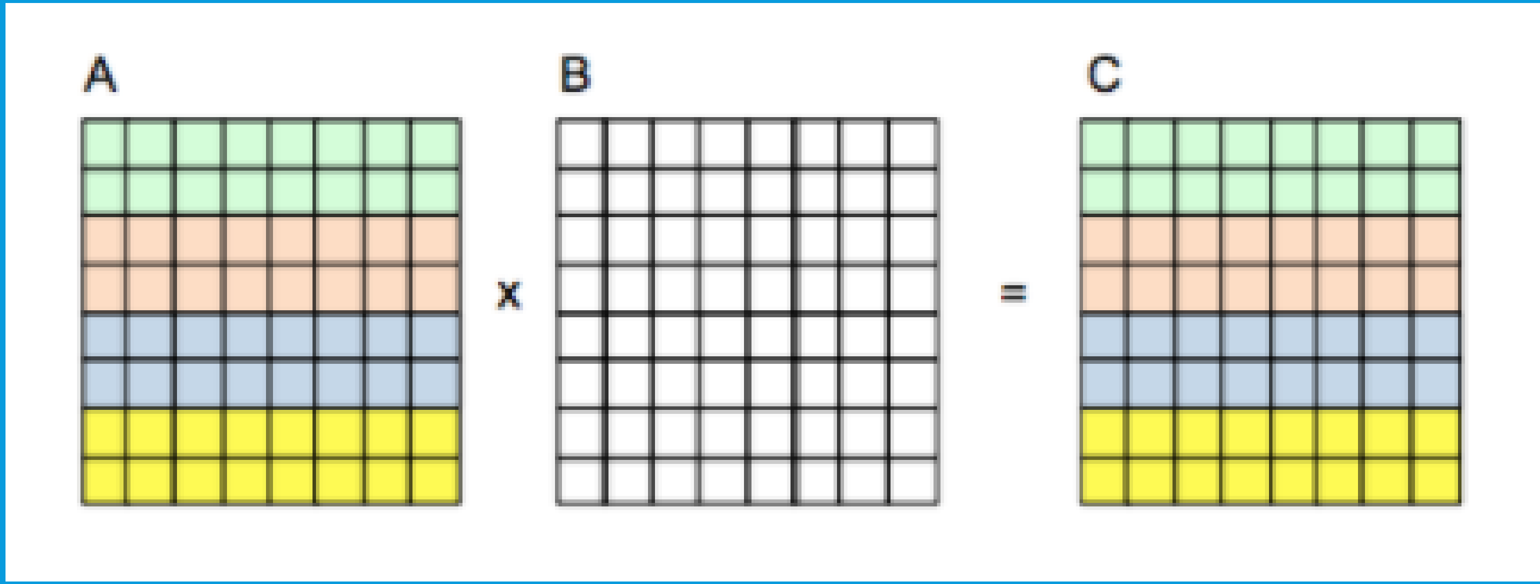
- Domain decomposition:



Block (left) and Cyclic (right) d.d.

DATA DECOMPOSITION

- Domain decomposition:



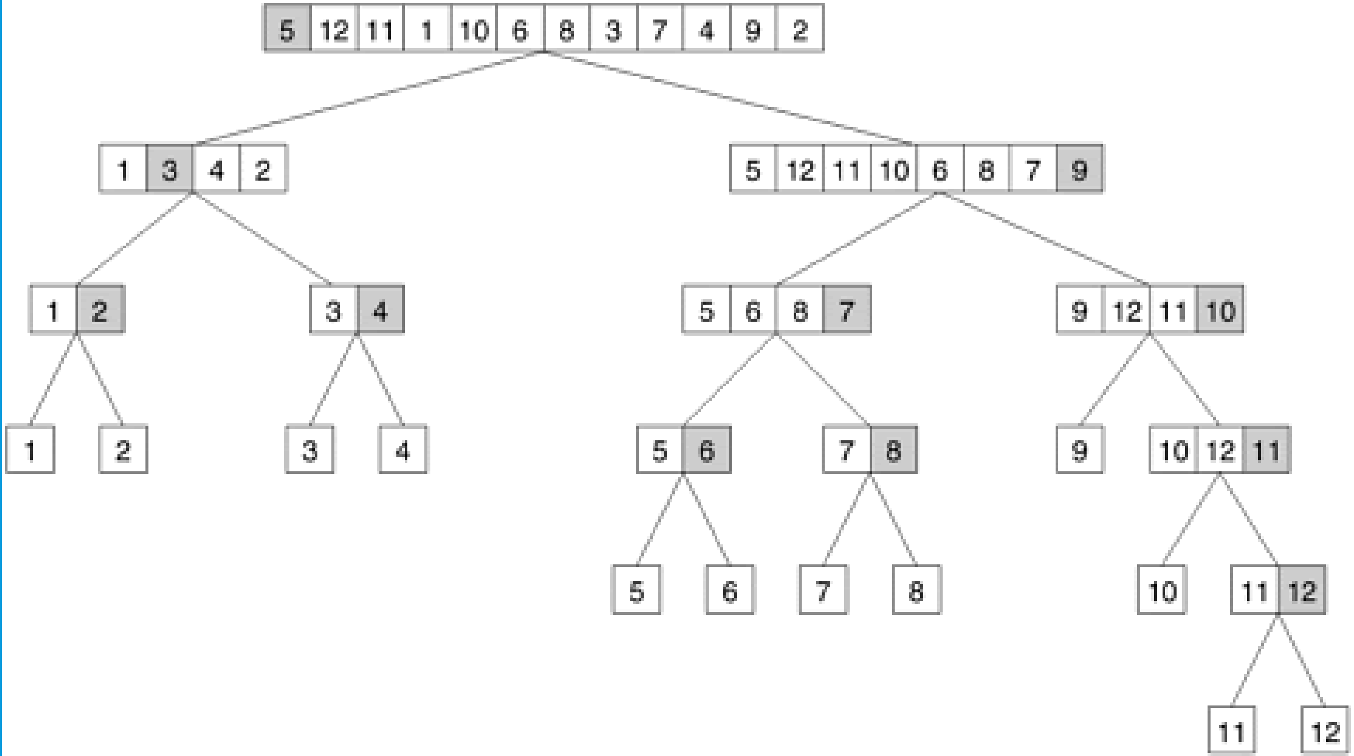
A and C partitioned by rows on 4 processors B is replicated.

DATA DECOMPOSITION

- Recursive Decomposition
  - Generally suited to problems that are solved using the divide-and-conquer strategy.
  - A given problem is decomposed into a set of sub-problems.
  - The sub-problems are recursively decomposed further until a desired granularity is reached.

DATA DECOMPOSITION

- A classic example of a divide-and-conquer algorithm is Quicksort.



DATA DECOMPOSITION

- What part do I need?

a11	a12	a13
a21	a22	a23
a31	a32	a33

x

b11	b12	b13
b21	b22	b23
b31	b32	b33

c11	c12	c13
c21	c22	c23
c31	c32	c33

Row from A and whole B

DATA DECOMPOSITION

- What part do I need?

a11	a12	a13		b11	b12	b13		c11	c12	c13
a21	a22	a23	x	b21	b22	b23	➡	c21	c22	c23
a31	a32	a33		b31	b32	b33		c31	c32	c33

Whole A and column from B

DATA DECOMPOSITION

- What part do I need?

a11	a12	a13
a21	a22	a23
a31	a32	a33

x

b11	b12	b13
b21	b22	b23
b31	b32	b33

c11	c12	c13
c21	c22	c23
c31	c32	c33

Whole A whole B

DATA DECOMPOSITION

- What part do I need?

a11	a12	a13
a21	a22	a23
a31	a32	a33

x

b11	b12	b13
b21	b22	b23
b31	b32	b33

→

c11	c12	c13
c21	c22	c23
c31	c32	c33

Yellow need a1*,a2* and b*1,b*2