

# Lecture 4: Principles of Parallel Algorithm Design (part 4)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Mapping Technique for Load Balancing

- Sources of overheads:
  - Inter-process interaction
  - Idling
- Goals to achieve:
  - To reduce interaction time
  - To reduce total amount of time some processes being idle
  - Remark: these two goals often conflict
- Classes of mapping:
  - Static
  - Dynamic

# Schemes for Static Mapping

- Mapping Based on Data Partitioning
- Task Graph Partitioning
- Hybrid Strategies

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Mapping Based on Data Partitioning

- By owner-computes rule, mapping the relevant data onto processes is equivalent to mapping tasks onto processes
- Array or Matrices
  - Block distributions
  - Cyclic and block cyclic distributions
- Irregular Data
  - Example: data associated with unstructured mesh
  - Graph partitioning

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# 1D Block Distribution

Example. Distribute rows or columns of matrix to different processes

row-wise distribution

$P_0$
$P_1$
$P_2$
$P_3$
$P_4$
$P_5$
$P_6$
$P_7$

column-wise distribution

$P_0$	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Multi-D Block Distribution

Example. Distribute blocks of matrix to different processes

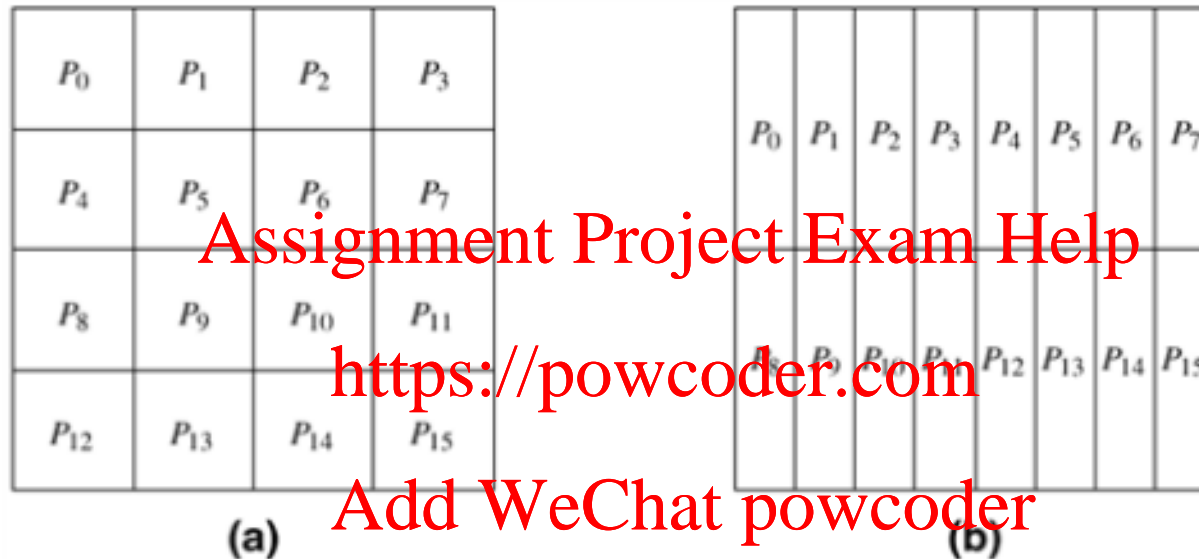


Figure 3.25. Examples of two-dimensional distributions of an array, (a) on a  $4 \times 4$  process grid, and (b) on a  $2 \times 8$  process grid.

# Load-Balance for Block Distribution

Example.  $n \times n$  dense matrix multiplication  $C = A \times B$  using  $p$  processes

- Decomposition based on output data.
- Each entry of  $C$  uses the same amount of computation.
- Either 1D or 2D block distribution can be used:
  - 1D distribution:  $\frac{n}{p}$  rows are assigned to a process
  - 2D distribution:  $n/\sqrt{p} \times n/\sqrt{p}$  size block is assigned to a process
- Multi-D distribution allows higher degree of concurrency.
- Multi-D distribution can also help to reduce interactions



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



(b)

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.



# Cyclic and Block Cyclic Distributions

- If the amount of work differs for different entries of a matrix, a block distribution can lead to load imbalances.
- Example. Doolittle's method of LU factorization of dense matrix

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

## Doolittle's method of LU factorization

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = LU = \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix}$$

By matrix-matrix multiplication

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

$$u_{1j} = a_{1j}, \quad j = 1, 2, \dots, n \text{ (1st row of } U)$$

$$l_{j1} = a_{j1}/u_{11}, \quad j = 1, 2, \dots, n \text{ (1st column of } L)$$

**For**  $i = 2, 3, \dots, n - 1$  **do**

$$u_{ii} = a_{ii} - \sum_{t=1}^{i-1} l_{it} u_{tj}$$

$$u_{ij} = a_{ij} - \sum_{t=1}^{i-1} l_{it} u_{tj} \quad \text{for } j = i + 1, \dots, n \text{ (ith row of } U)$$

$$l_{ji} = \frac{a_{ji} - \sum_{t=1}^{i-1} l_{jt} u_{ti}}{u_{ii}} \quad \text{for } j = i + 1, \dots, n \text{ (ith column of } L)$$

**End**

$$u_{nn} = a_{nn} - \sum_{t=1}^{n-1} l_{nt} u_{tn}$$

# Serial Column-Based LU

```
1.  procedure COL_LU (A)
2.  begin
3.      for k := 1 to n do
4.          for j := k to n do
5.              A[j, k] := A[j, k]/A[k, k];
6.          endfor;
7.          for i := k + 1 to n do
8.              for j := k + 1 to n do
9.                  A[i, j] := A[i, j] - A[i, k] × A[k, j];
10.             endfor;
11.          endfor;
12.      /*
13.      After this iteration, column A[k + 1 : n, k] is logically the kth
14.      column of L and row A[k, k : n] is logically the kth row of U.
15.      */
16.  end COL_LU
```

- Remark: Matrices L and U share space with A

# Work used to compute Entries of L and U

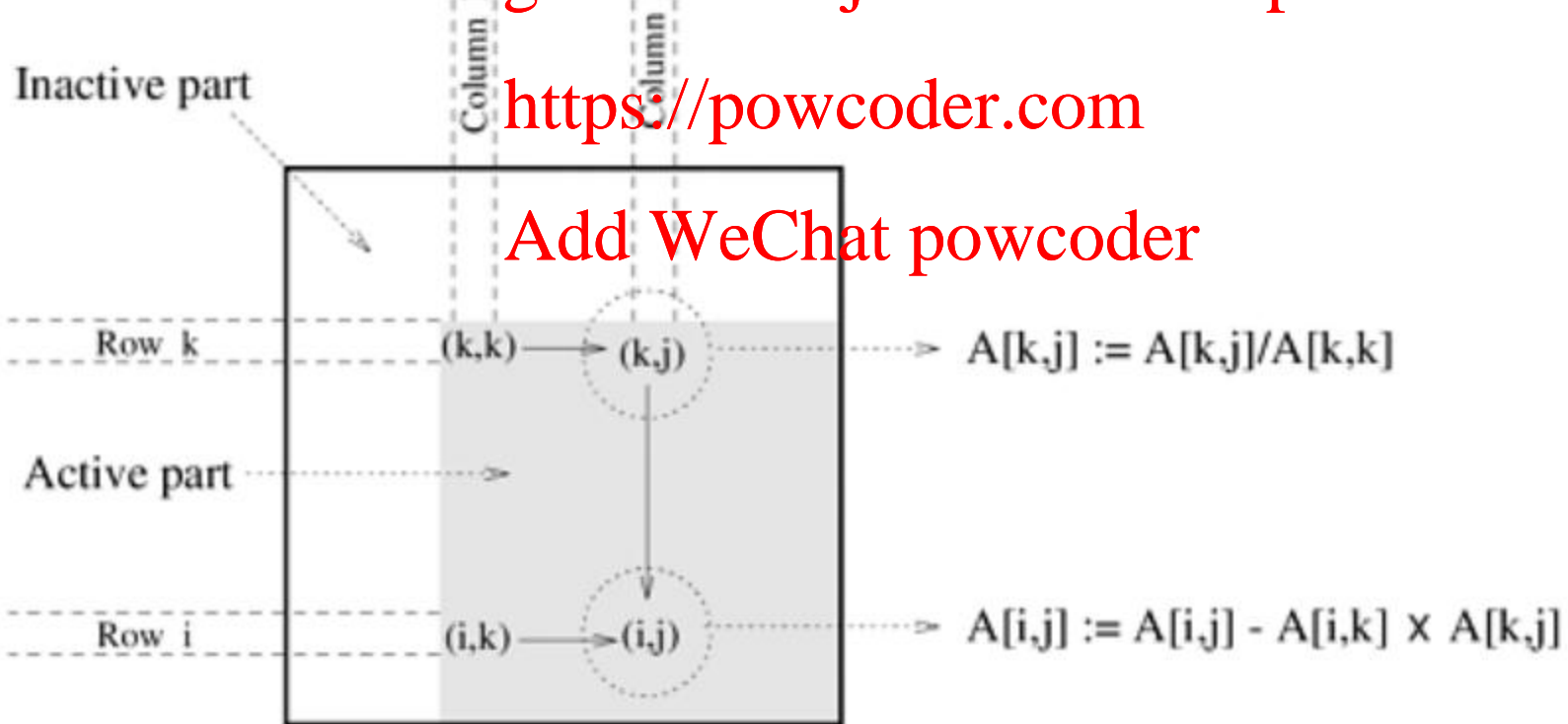
$$\begin{pmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix} \rightarrow \begin{pmatrix} L_{1,1} & 0 & 0 \\ L_{2,1} & L_{2,2} & 0 \\ L_{3,1} & L_{3,2} & L_{3,3} \end{pmatrix} \cdot \begin{pmatrix} U_{1,1} & U_{1,2} & U_{1,3} \\ 0 & U_{2,2} & U_{2,3} \\ 0 & 0 & U_{3,3} \end{pmatrix}$$

1: $A_{1,1} \rightarrow L_{1,1}U_{1,1}$	6: $A_{2,2} = A_{2,2} - L_{2,1}U_{1,2}$	11: $L_{3,2} = A_{3,2}U_{2,2}^{-1}$
2: $L_{2,1} = A_{2,1}U_{1,1}^{-1}$	7: $A_{3,2} = A_{3,2} - L_{3,1}U_{1,2}$	12: $U_{2,3} = L_{2,2}^{-1}A_{2,3}$
3: $L_{3,1} = A_{3,1}U_{1,1}^{-1}$	8: $A_{2,3} = A_{2,3} - L_{2,1}U_{1,3}$	<del>13: <math>A_{3,3} = A_{3,3} - L_{3,2}U_{2,3}</math></del>
4: $U_{1,2} = L_{1,1}^{-1}A_{1,2}$	9: <del><math>A_{3,3} = A_{3,3} - L_{3,1}U_{1,3}</math></del>	<del>14: <math>A_{3,3} \rightarrow L_{3,3}U_{3,3}</math></del>
5: $U_{1,3} = L_{1,1}^{-1}A_{1,3}$	10: $A_{2,2} \rightarrow L_{2,2}U_{2,2}$	

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



3.28. A typical computation in Gaussian elimination and the active part of the coefficient matrix during the  $k$ th iteration of the outer loop.

- Block distribution of LU factorization tasks leads to load imbalance.

<b>P<sub>0</sub></b> T <sub>1</sub>	<b>P<sub>3</sub></b> T <sub>4</sub>	<b>P<sub>6</sub></b> T <sub>5</sub>
<b>P<sub>1</sub></b> T <sub>2</sub>	<b>P<sub>4</sub></b> T <sub>6</sub> T <sub>10</sub>	<b>P<sub>7</sub></b> T <sub>8</sub> T <sub>12</sub>
<b>P<sub>2</sub></b> T <sub>3</sub>	<b>P<sub>5</sub></b> T <sub>7</sub> T <sub>11</sub>	<b>P<sub>8</sub></b> T <sub>9</sub> T <sub>13</sub> T <sub>14</sub>

Assignment Project Exam Help

<https://powcoder.com>

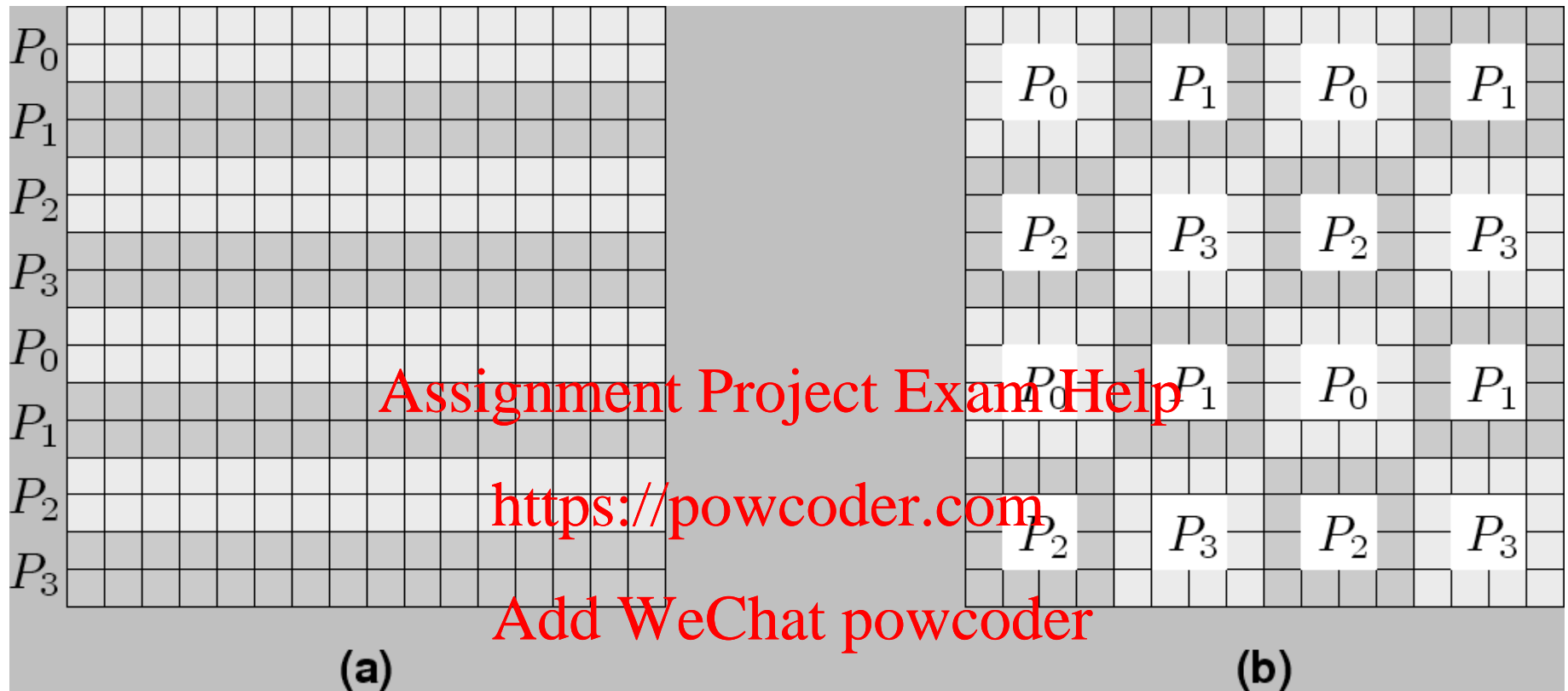
Add WeChat powcoder

# Block-Cyclic Distribution

- A variation of block distribution that can be used to alleviate the load-imbalance.

Assignment Project Exam Help

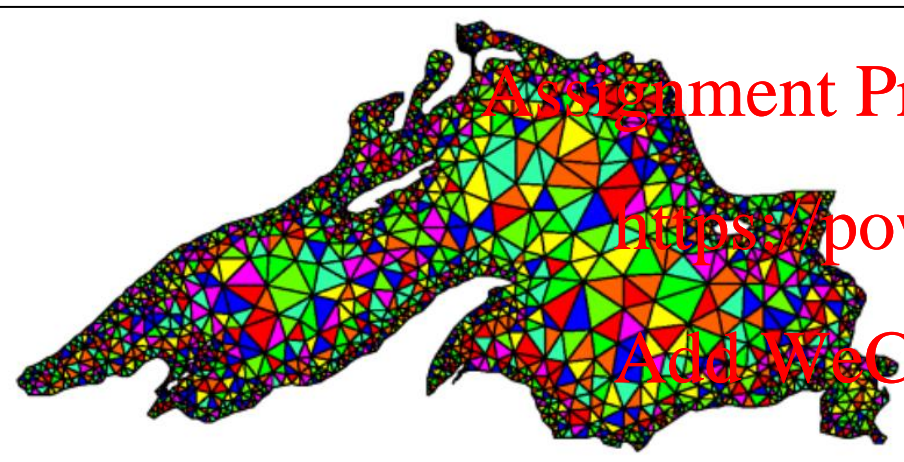
- Steps <https://powcoder.com>
  1. Partition an array into many more blocks than the number of available processes
  2. Assign blocks to processes in a round-robin manner so that each process gets several non-adjacent blocks.



- (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 4x4, and it is mapped onto a 2x2 grid of processes in a wraparound fashion.
- **Cyclic distribution:** when the block size =1

# Graph Partitioning

- Assign equal number of nodes (or cells) to each process
- Minimize edge count of the graph partition



Random Partitioning



Partitioning for Minimizing Edge-Count

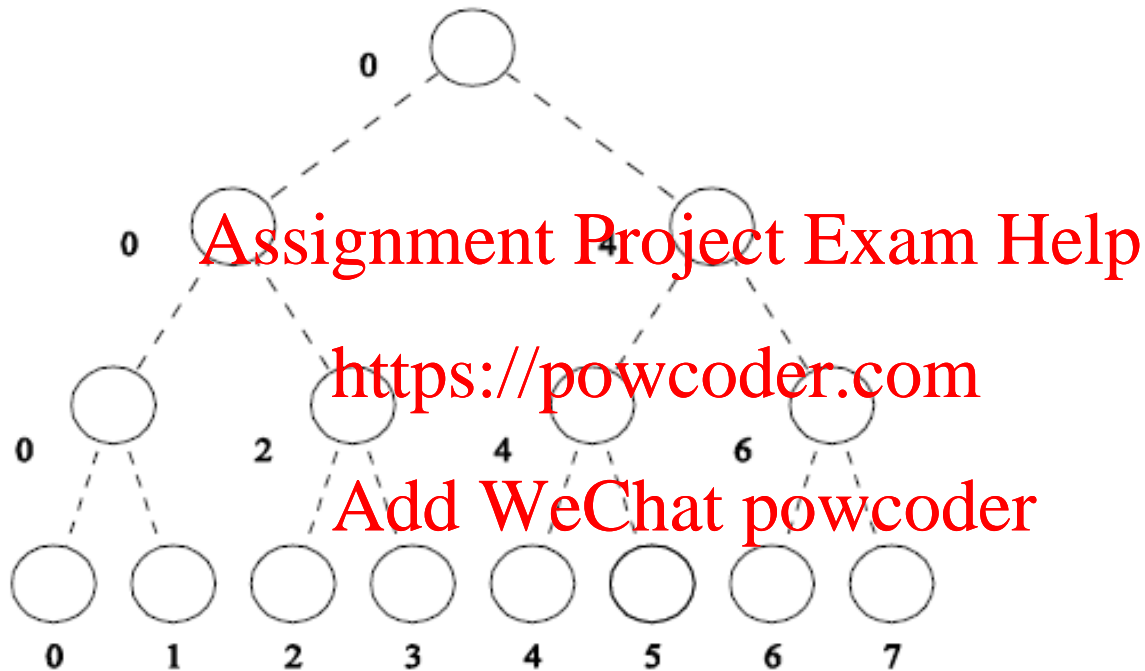


# Mappings Based on Task Partitioning

- Mapping based on task partitioning can be used when computation is naturally expressed in the form of a static task-dependency graph with known sizes. <https://powcoder.com>
- Finding optimal mapping minimizing idle time and minimizing interaction time is NP-complete
- Heuristic solutions exist for many structured graphs

# Mapping a Binary Tree Task-Dependency Graph

- Finding min.



- Mapping the tree graph onto 8 processes
- Mapping minimizes the interaction overhead by mapping independent tasks onto the same process (i.e., process 0) and others on processes only one communication link away from each other
- Idling exists. This is inherent in the graph

# Mapping a Sparse Graph

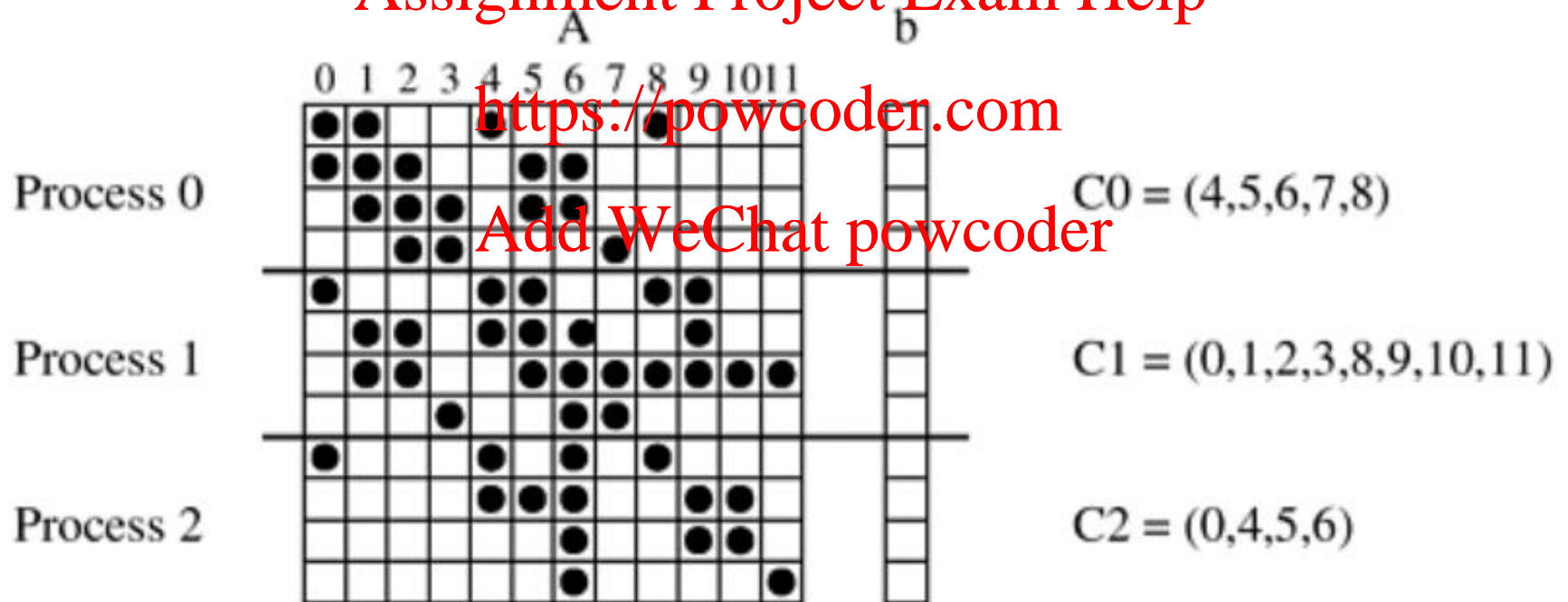
Example. Sparse matrix-vector multiplication using 3 processes

- Arrow distribution

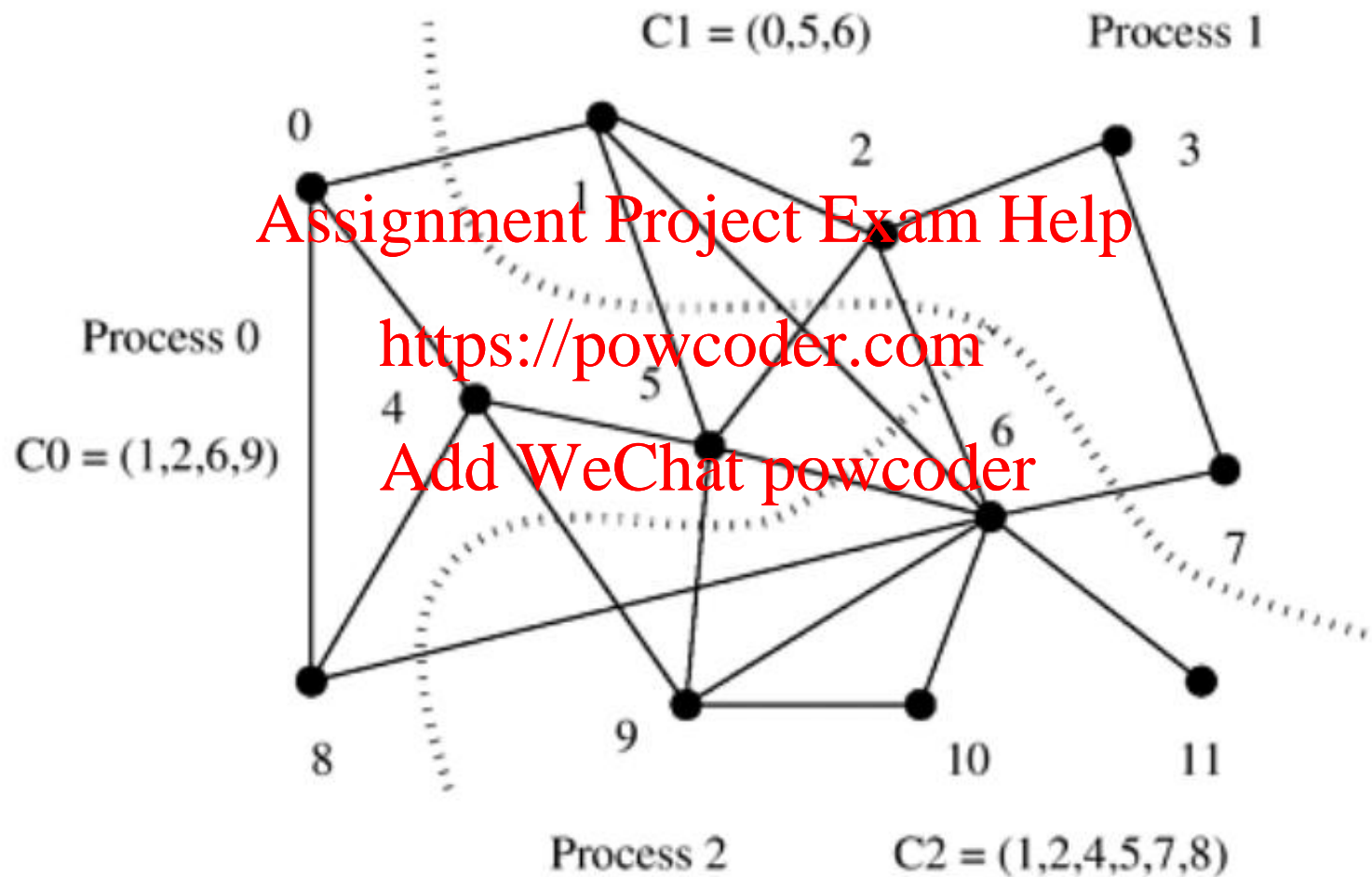
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



- Partitioning task interaction graph to reduce interaction overhead



# Schemes for Dynamic Mapping

- When static mapping results in highly imbalanced distribution of work among processes or when task-dependency graph is dynamic, use dynamic mapping
- Primary goal is to balance load – dynamic load balancing
  - Example: Dynamic load balancing for AMR
- Types
  - Centralized
  - Distributed

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Centralized Dynamic Mapping

- Processes
  - Master: manage a group of available tasks
  - Slave: depend on master to obtain work
- Idea
  - When a slave process has no work, it takes a portion of available work from master
  - When a new task is generated, it is added to the pool of tasks in the master process
- Potential problem
  - When many processes are used, master process may become bottleneck
- Solution
  - Chunk scheduling: every time a process runs out of work it gets a group of tasks.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Distributed Dynamic Mapping

- All processes are peers. Tasks are distributed among processes which exchange tasks at run time to balance work
- Each process can send or receive work from other processes
  - How are sending and receiving processes paired together
  - Is the work transfer initiated by the sender or the receiver?
  - How much work is transferred?
  - When is the work transfer performed?

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Techniques to Minimize Interaction Overheads

- Maximize data locality
  - Maximize the reuse of recently accessed data
  - Minimize volume of data-exchange
    - Use high dimensional distribution. Example: 2D block distribution for matrix multiplication
  - Minimize frequency of interactions
    - Reconstruct algorithm such that shared data are accessed and used in large pieces.
    - Combine messages between the same source-destination pair

Assignment Project Exam Help

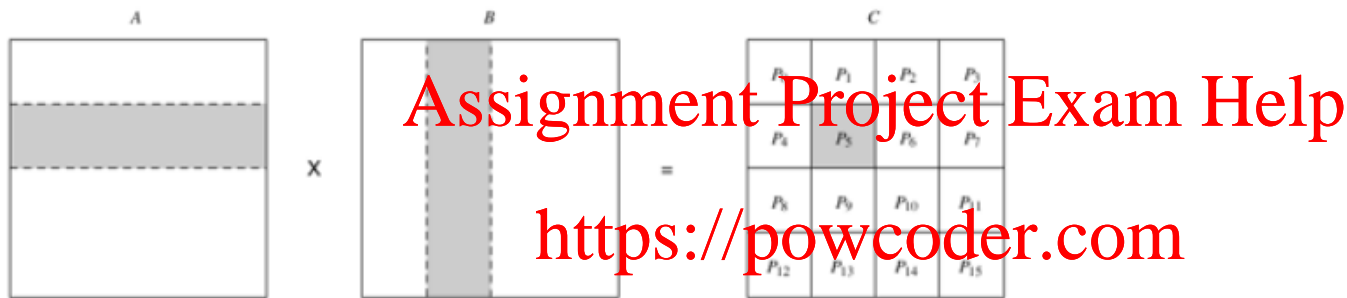
<https://powcoder.com>

Add WeChat powcoder



# Techniques to Minimize Interaction Overheads

- Minimize contention and hot spots
  - Contention occur when multi-tasks try to access the same resources concurrently: multiple processes sending message to the same process; multiple simultaneous accesses to the same memory block



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- Using  $C_{i,j} = \sum_{k=0}^{\sqrt{p}-1} A_{i,k} B_{k,j}$  causes contention. For example,  $C_{0,0}$ ,  $C_{0,1}$ ,  $C_{0,\sqrt{p}-1}$  attempt to read  $A_{0,0}$ , at once.
- A contention-free manner is to use:

$$C_{i,j} = \sum_{k=0}^{\sqrt{p}-1} A_{i,(i+j+k)\% \sqrt{p}} B_{(i+j+k)\% \sqrt{p},j}$$

All tasks  $P_{*,j}$  that work on the same row of C access block  $A_{i,(i+j+k)\% \sqrt{p}}$ , which is different for each task.

# Techniques to Minimize Interaction Overheads

- Overlap computations with interactions
  - Use non-blocking communication
- Replicate data or computations
  - Replicate a copy of shared data on each process if possible, so that there is only initial interaction during replication.
- Use collective interaction operations
- Overlap interactions with other interactions

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Parallel Algorithm Models

- Data parallel
  - Each task performs similar operations on different data
  - Typically statically map tasks to processes
- Task graph
  - Use task dependency graph to promote locality or reduce interactions
- Master-slave
  - One or more master processes generating tasks
  - Allocate tasks to slave processes
  - Allocation may be static or dynamic
- Pipeline/producer-consumer
  - Pass a stream of data through a sequence of processes
  - Each performs some operation on it
- Hybrid
  - Apply multiple models hierarchically, or apply multiple models in sequence to different phases

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder