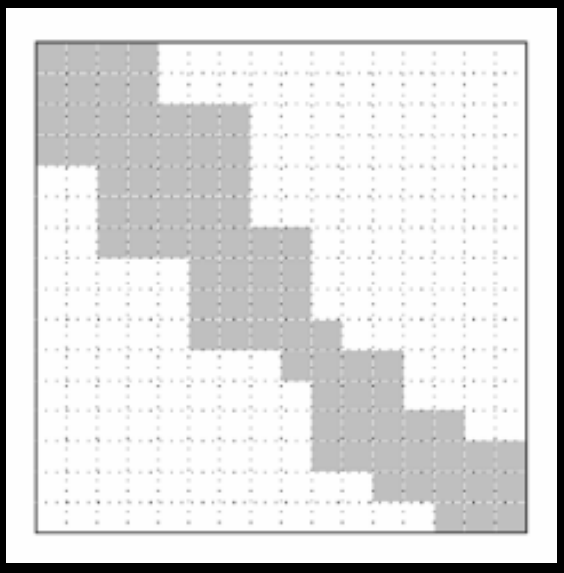
III. Randomized Block Distribution

Block-cyclic may not be able balance computations
→ Distribution of work has special patterns



p = 16
64

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}

(2d Block-cyclic)

⇒ Partition array into many more blocks than the available processes
Distribute blocks uniformly and randomly to the processes

e.g. for 1-d randomized block distribution with, some p and α
Define a vector V of length αp
$V[j] \leftarrow j$ for $0 \le j < \alpha p$
Randomly permute V.
Assign $V[i\alpha \text{ to } (i+1)\alpha - 1]$ to P_i

For $p = 4$, $\alpha = 3$

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

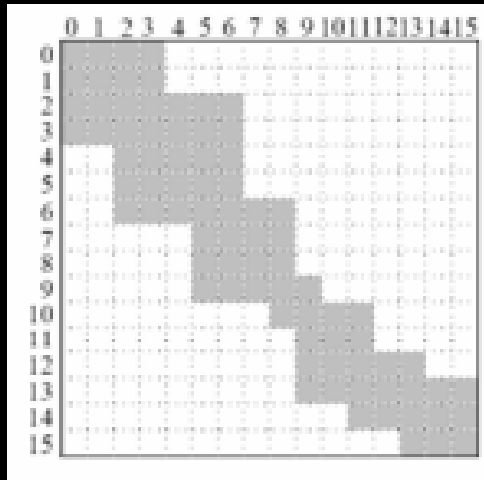
$\text{random}(V) = [8, 2, 6, 0, 3, 7, 11, 1, 9, 5, 4, 10]$

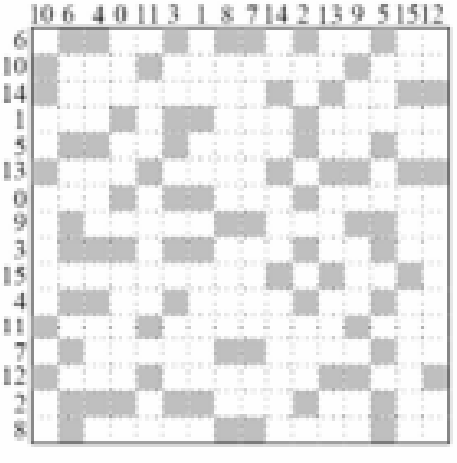
mapping = 8 2 6 | 0 3 7 | 11 1 9 | 5 4 10

P_0 P_1 P_2 P_3

2-d randomized block distribution

For an $n \times n$ array, p processes = $\sqrt{p} \times \sqrt{p}$, randomly permute 2 vectors of length $\alpha\sqrt{p}$ each and use them to choose the row and column indices of the blocks to be assigned to each process.





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

IV. Graph Partitioning

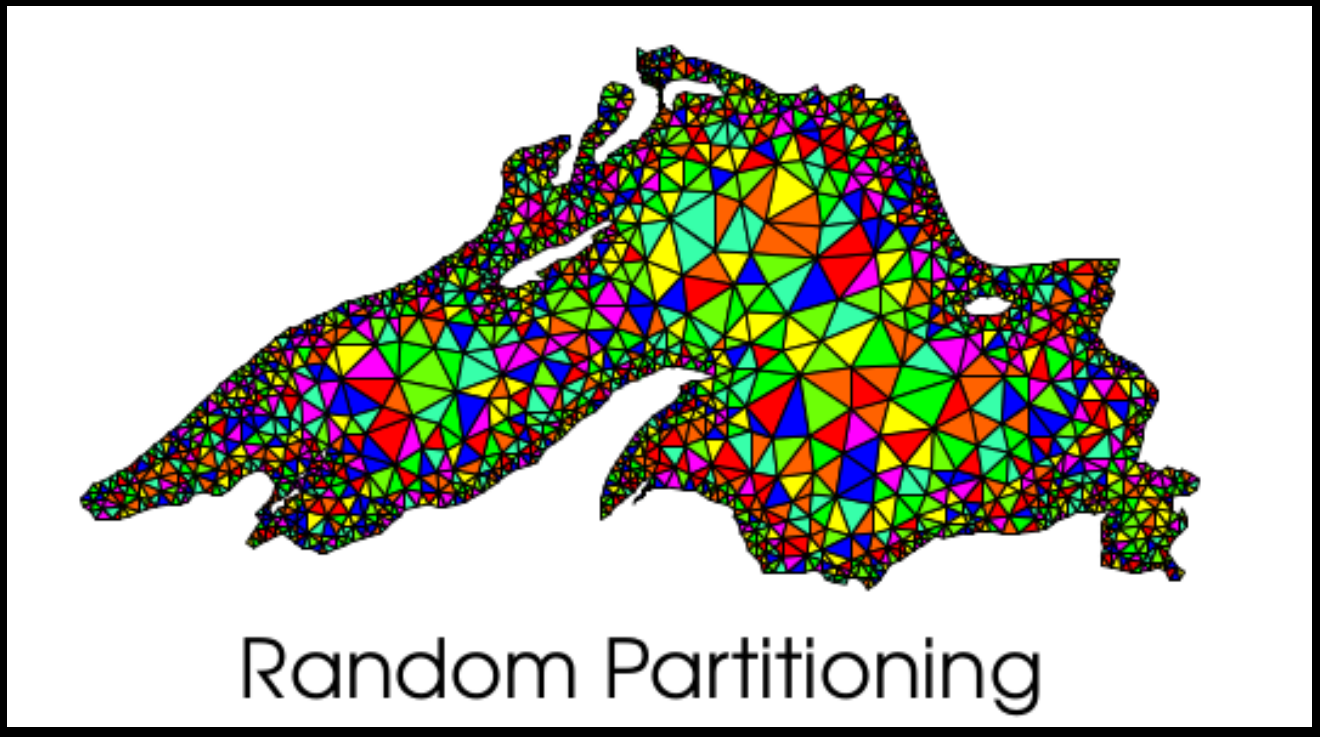
- All above 3 partitions are for arrays and matrices.

- What if interaction between Tasks is irregular?

e.g. Numerical Simulations of Physical phenomena

- Physical domain is discretized and represented by a mesh of elements
- Compute certain physical quantities representing the phenomena at each mesh point.
- Relate it to the computations at adjacent mesh points

Red
Orange
L. Blue
D. Blue
Magenta
Yellow
L. Green
D. Green



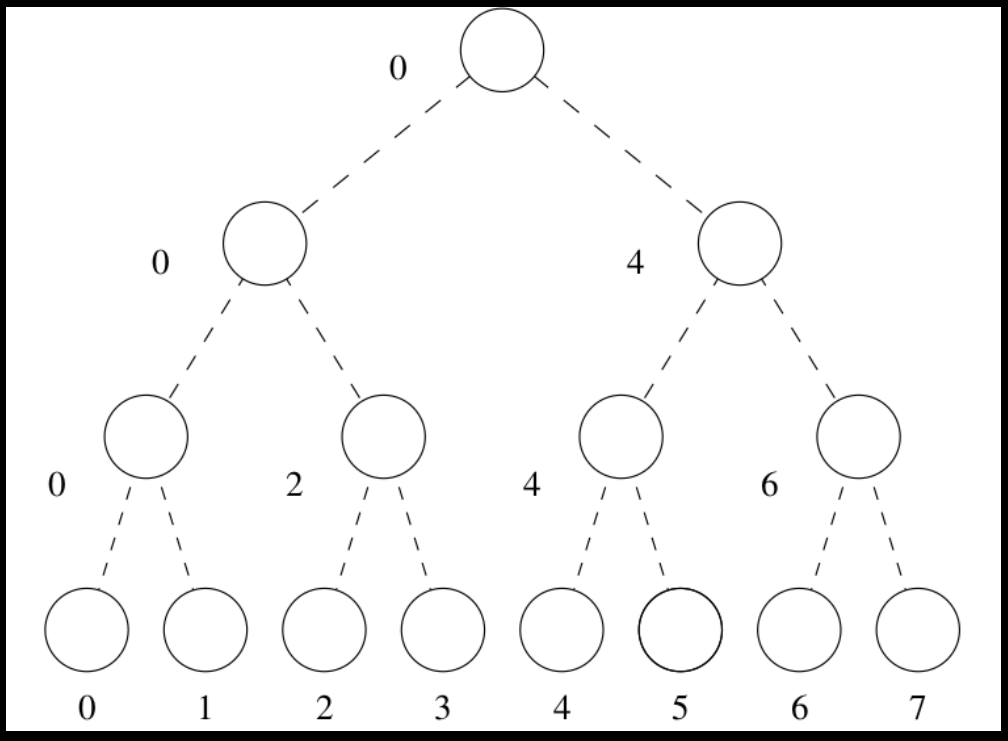
Good Load Balancing as same number of points are assigned to each process.

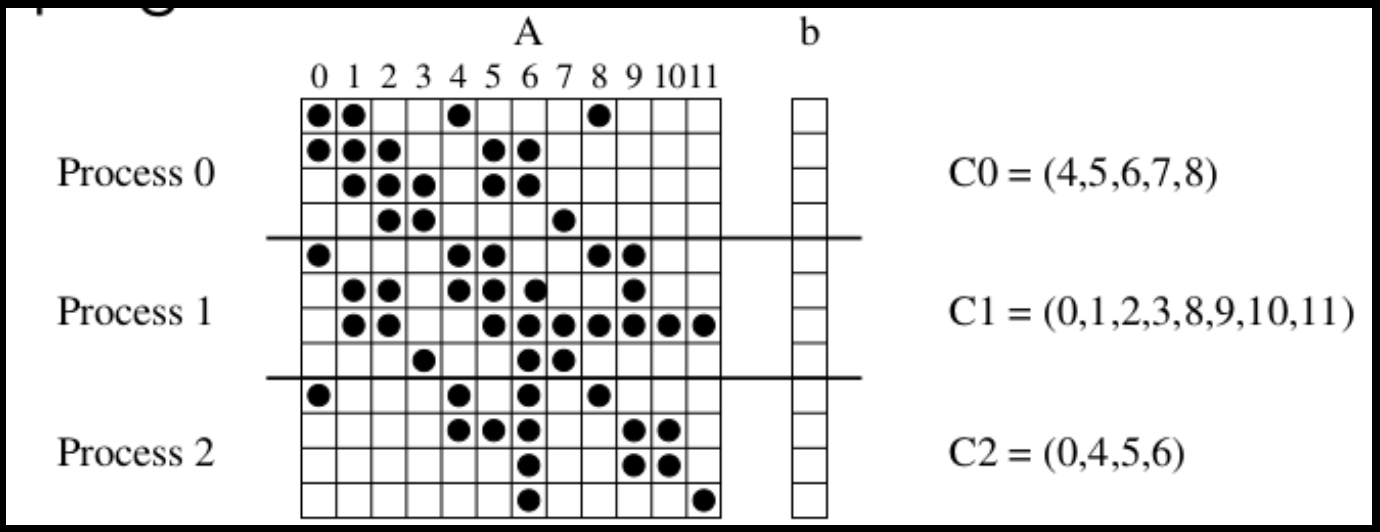
But what about Interaction Overhead?

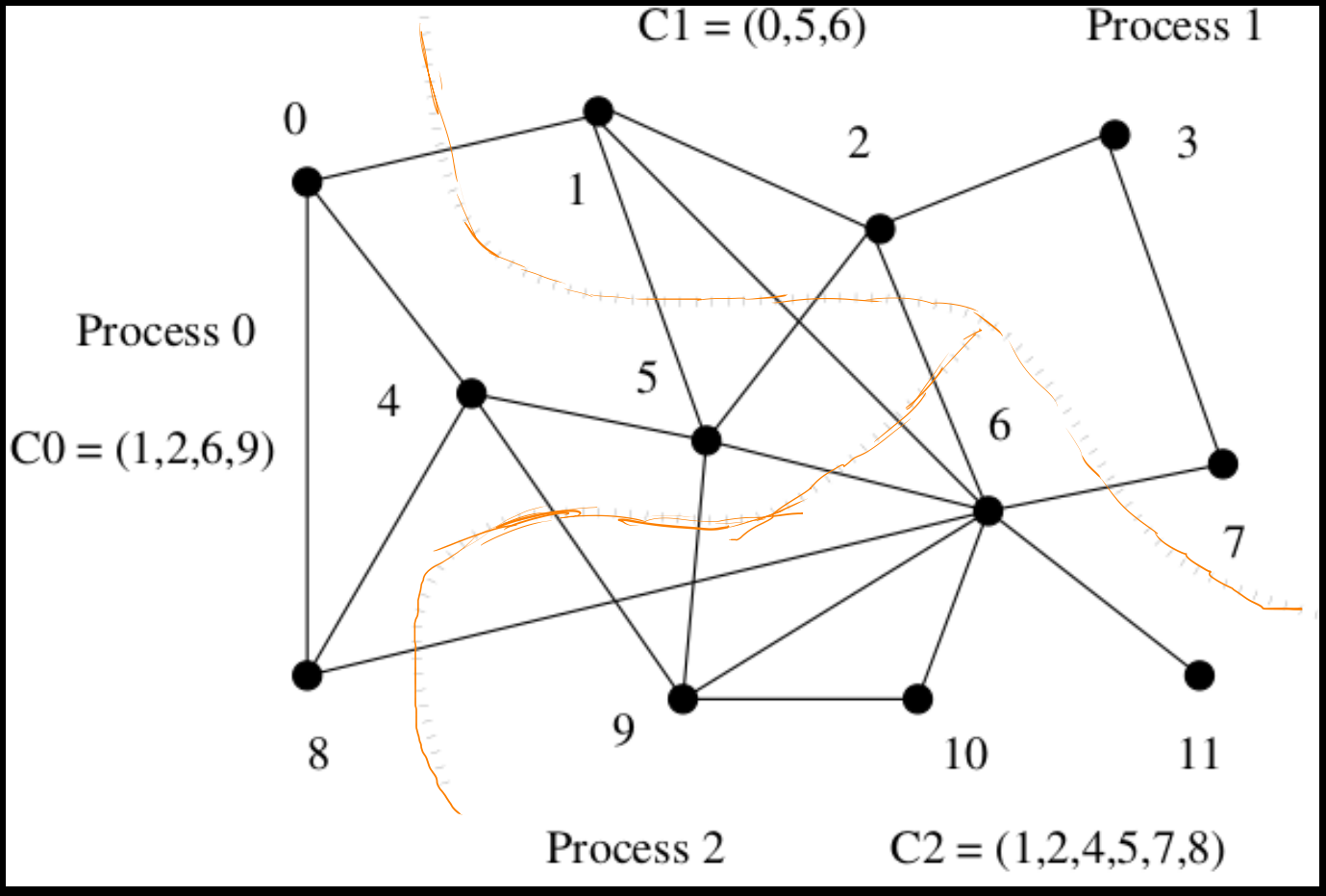
Partitioning for minimum edge-cut.

Mapping using Task Partitioning

- Task Dependency Graph
- Task Interaction graph

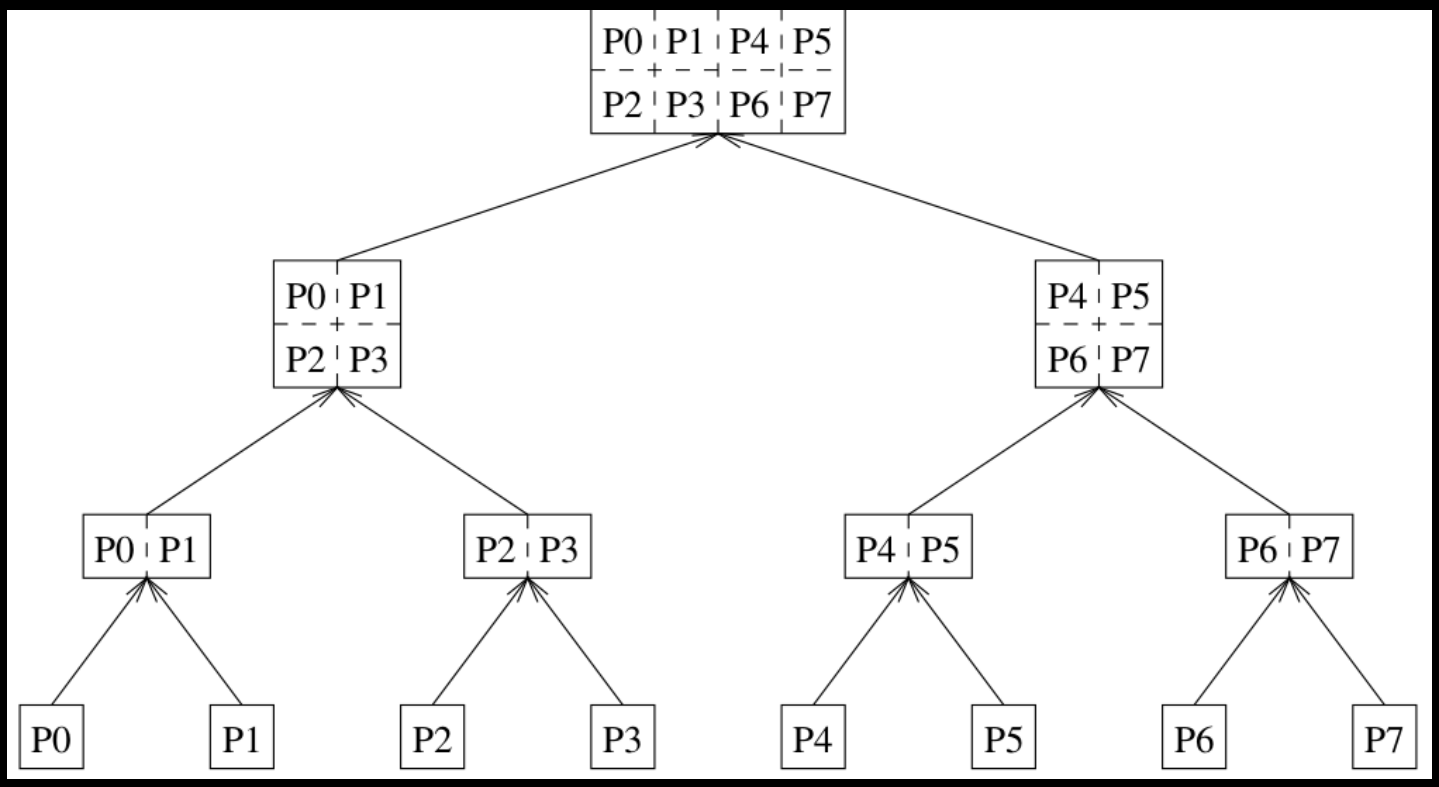






Hierarchical Mapping: Use different decompositions and mappings at different levels.

e.g: First level of Task partitioning
Data partitioning within each level


P0 P1 P4 P5
P2 P3 P6 P7
P0 P1
P2 P3
P4 P5
P6 P7
P0 P1
P2 P3
P4 P5
P6 P7
P0
P1
P2
P3
P4
P5
P6
P7