



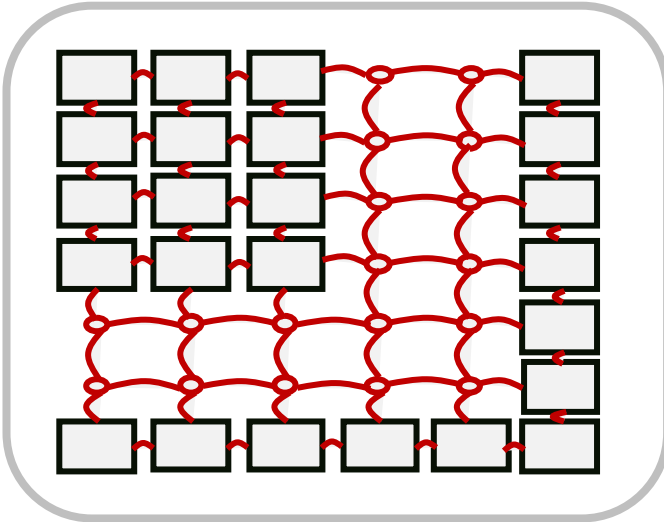
A Unified Two-Dimensional Design Space for the MPI All-to-All Collective

Presenter: Ke Fan

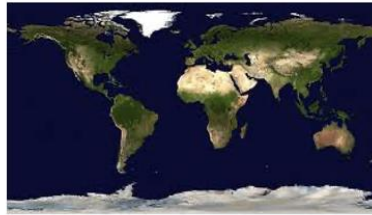
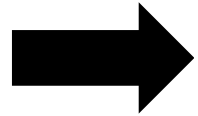
Authors: Ke Fan, Guimu Guo, and Sidharth Kumar



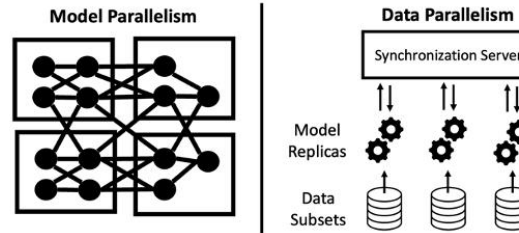
Exascale Computing Demands Effective Parallelism



Exascale Supercomputers



Earth and Climate Modeling



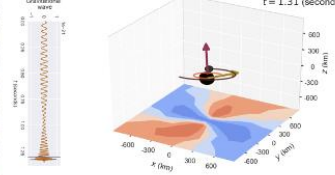
Parallel Machine Learning



**Inter-process Data
Movement**



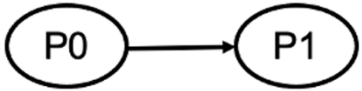
Colliding and Wobbling Black Holes



Black Holes Simulation

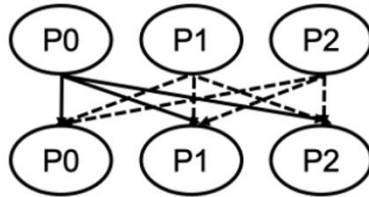
Three Fundamental Interfaces of Data Movement

Point-to-point

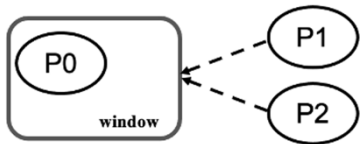


MPI_Send
MPI_Recv

Collective



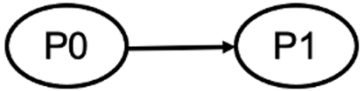
One-sided



MPI_Put
MPI_Get

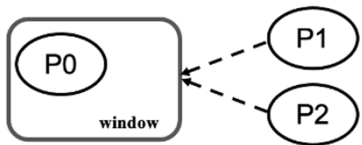
Collective Involves Communication Among All Processes

Point-to-point



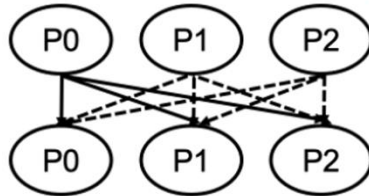
MPI_Send
MPI_Recv

One-sided

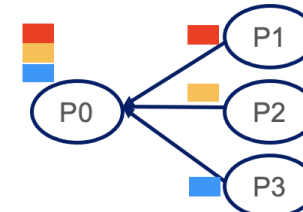


MPI_Put
MPI_Get

Collective

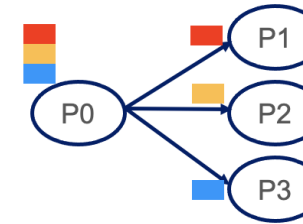


All-to-One



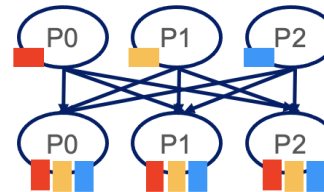
e.g., MPI_Reduce

One-to-All



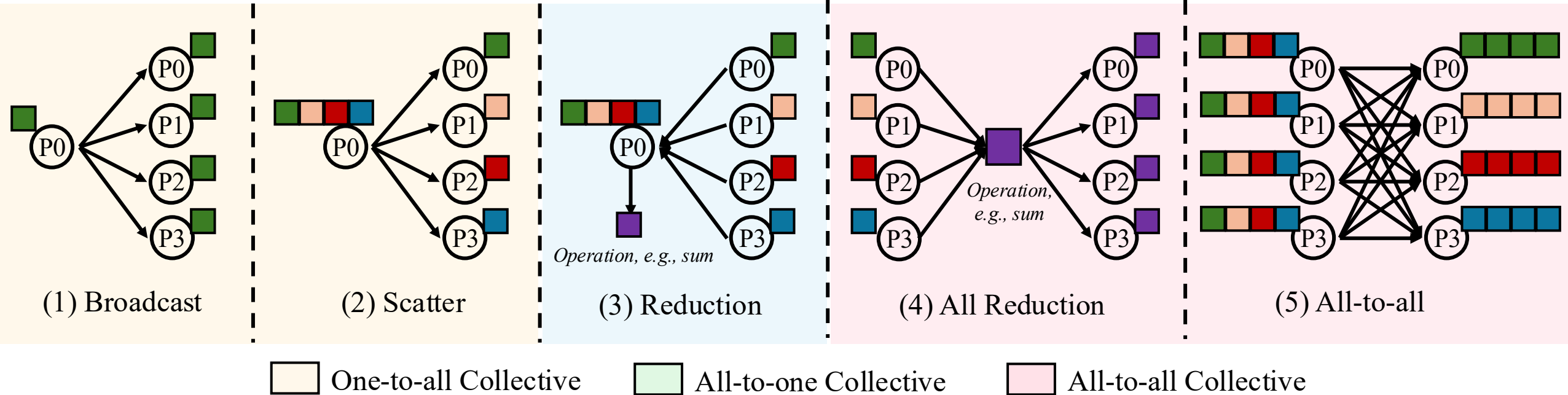
e.g., MPI_Scatter

All-to-All

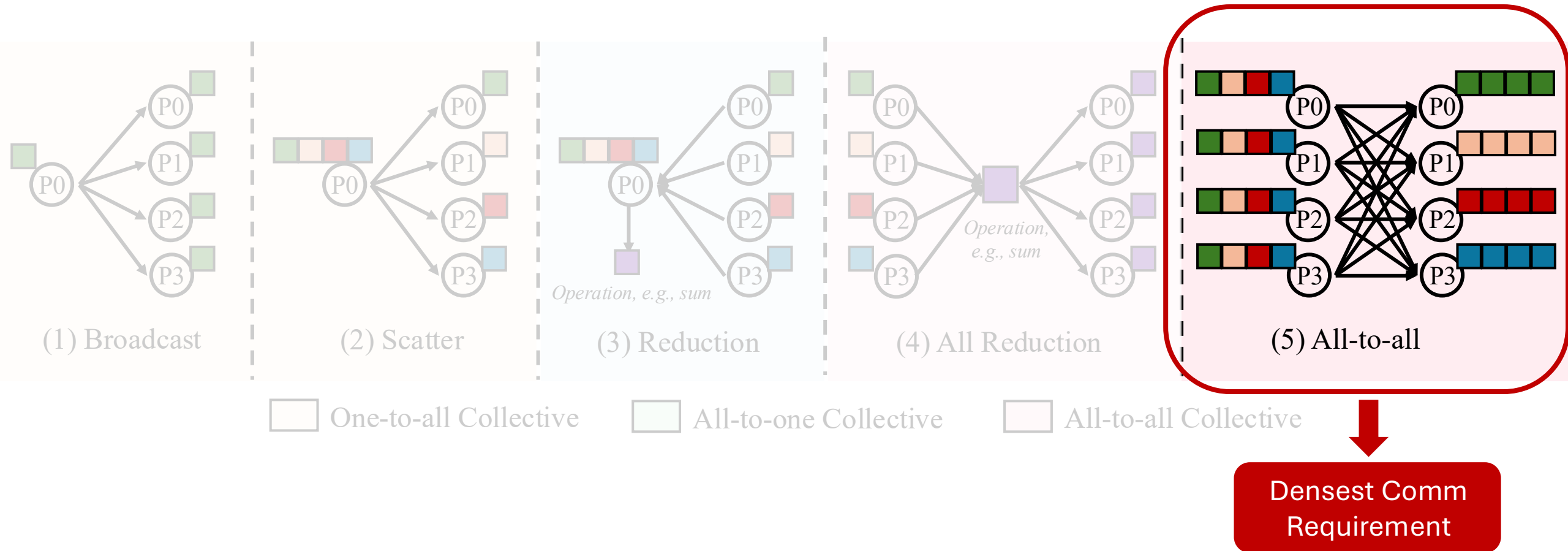


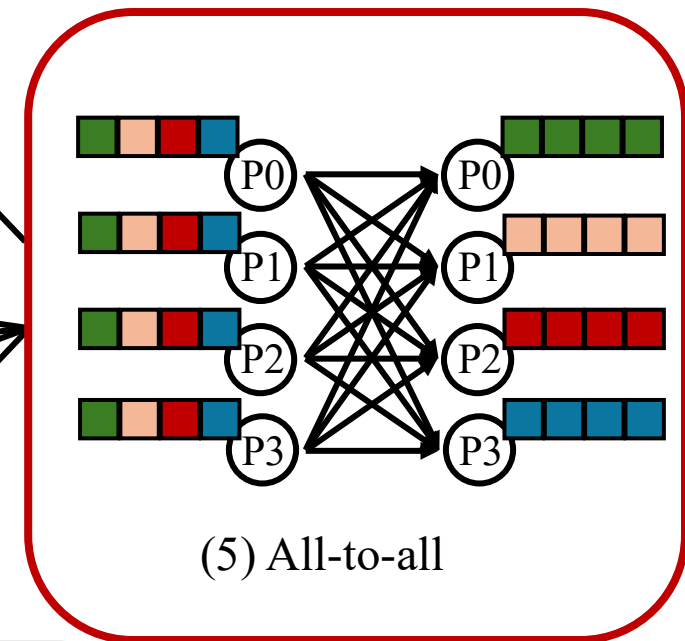
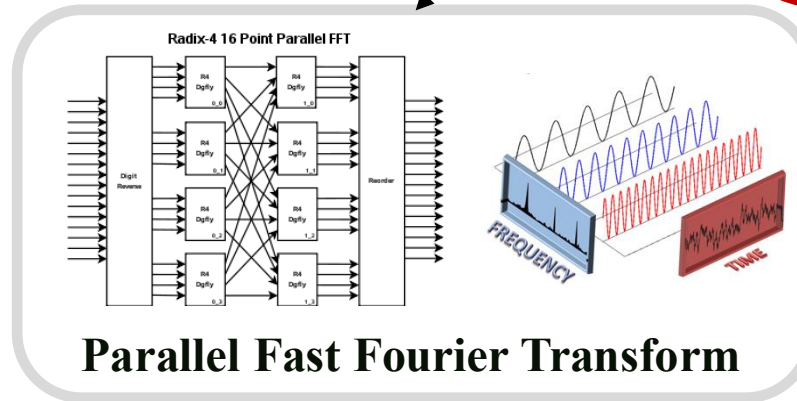
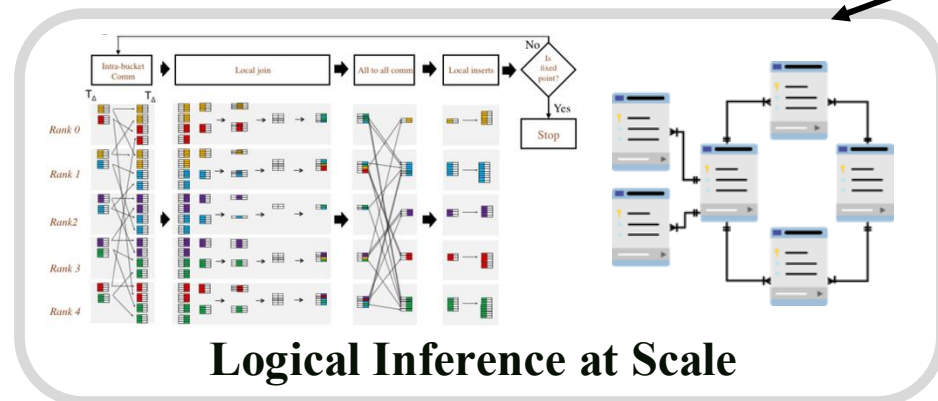
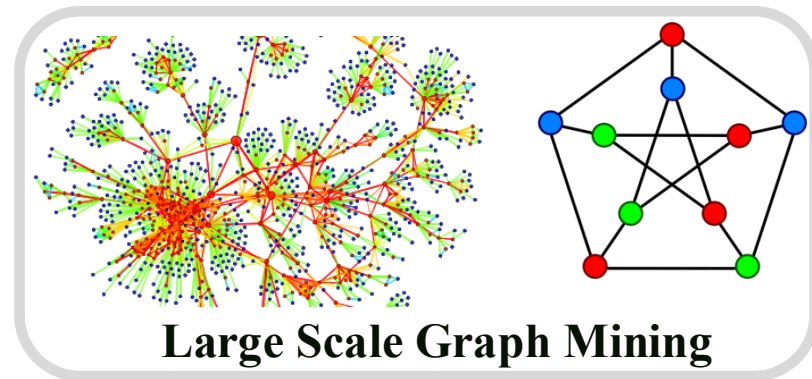
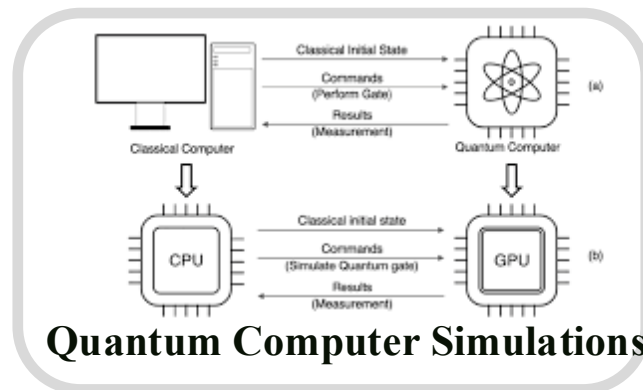
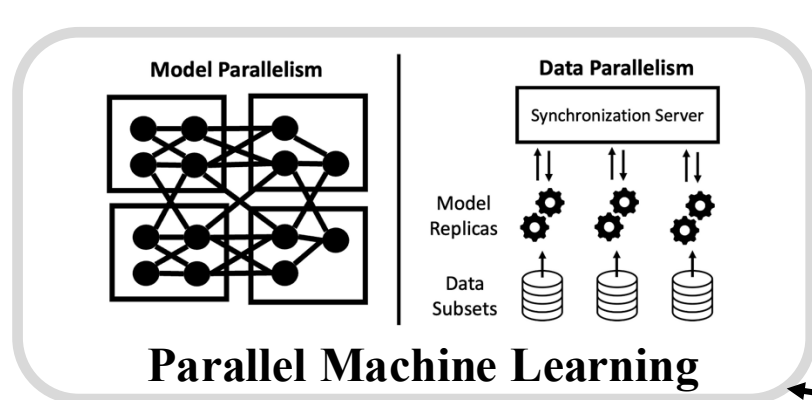
e.g., MPI_Alltoall

Some Widely Used Collective Primitives

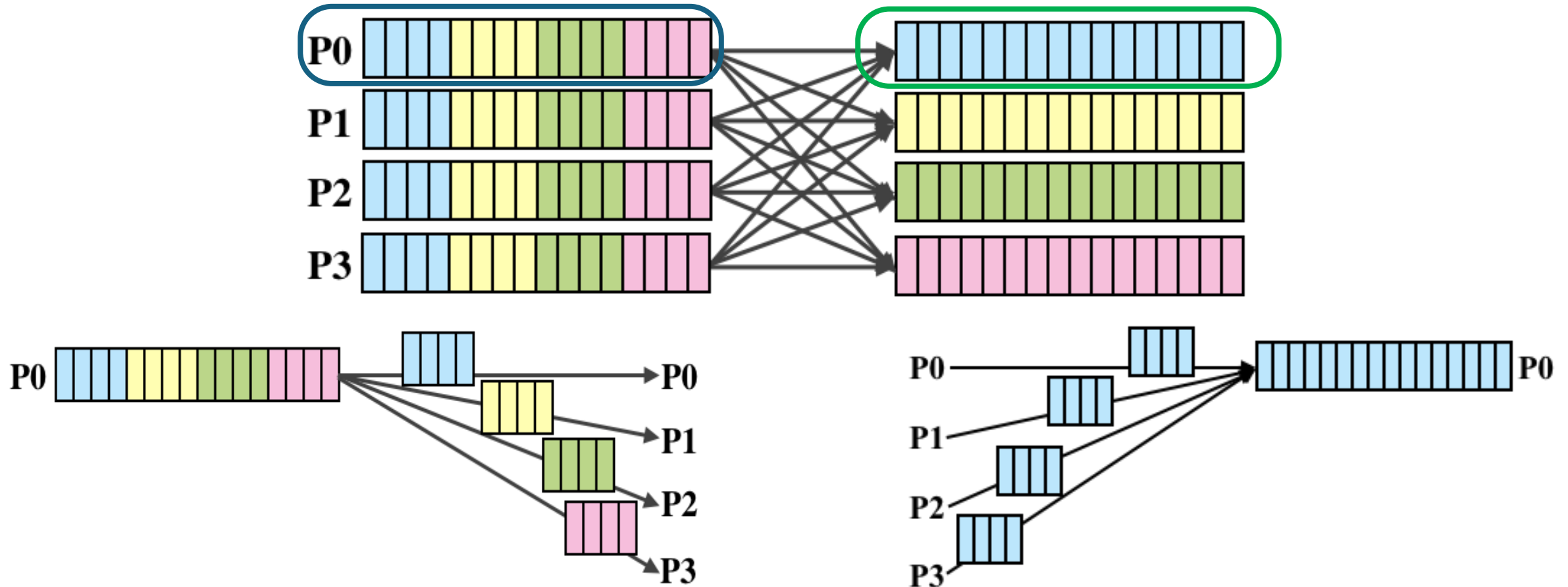


All-to-all Data Exchange is the Most Challenging to Scale and Optimize





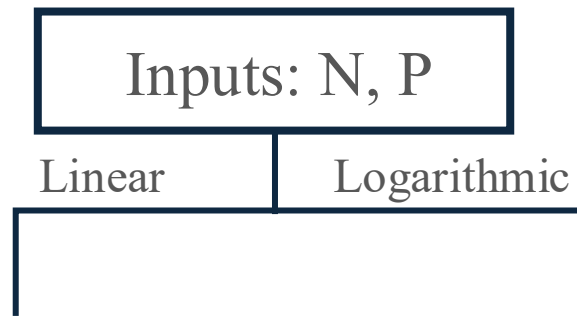
All-to-all – Every process Sends and receives data from every other process



Standard MPI All-to-all Implementations

N: Size of data-block

P: Process count



Standard MPI All-to-all Implementations: Spread-out

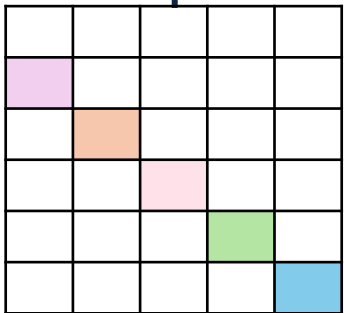
N: Size of data-block

P: Process count

Inputs: N, P

Linear

P0



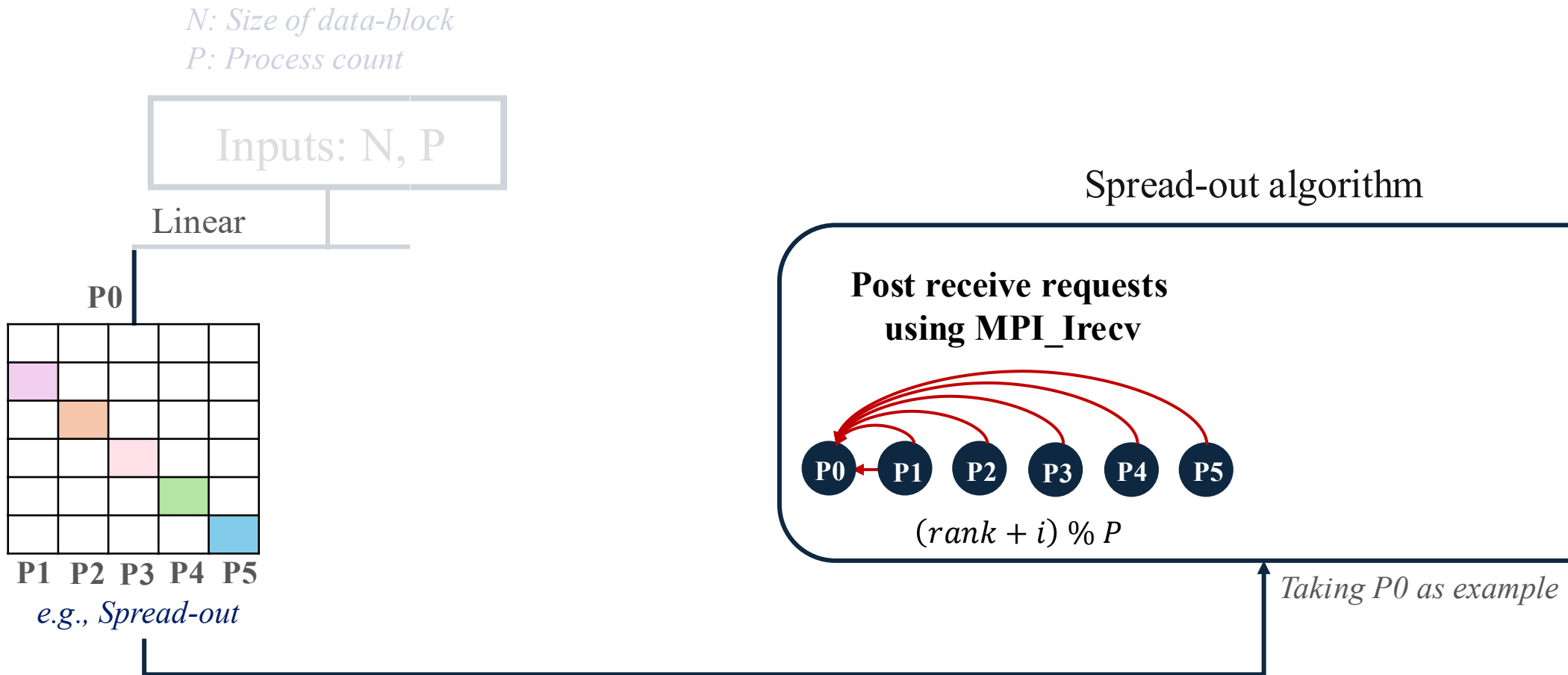
P1 P2 P3 P4 P5

e.g., Spread-out

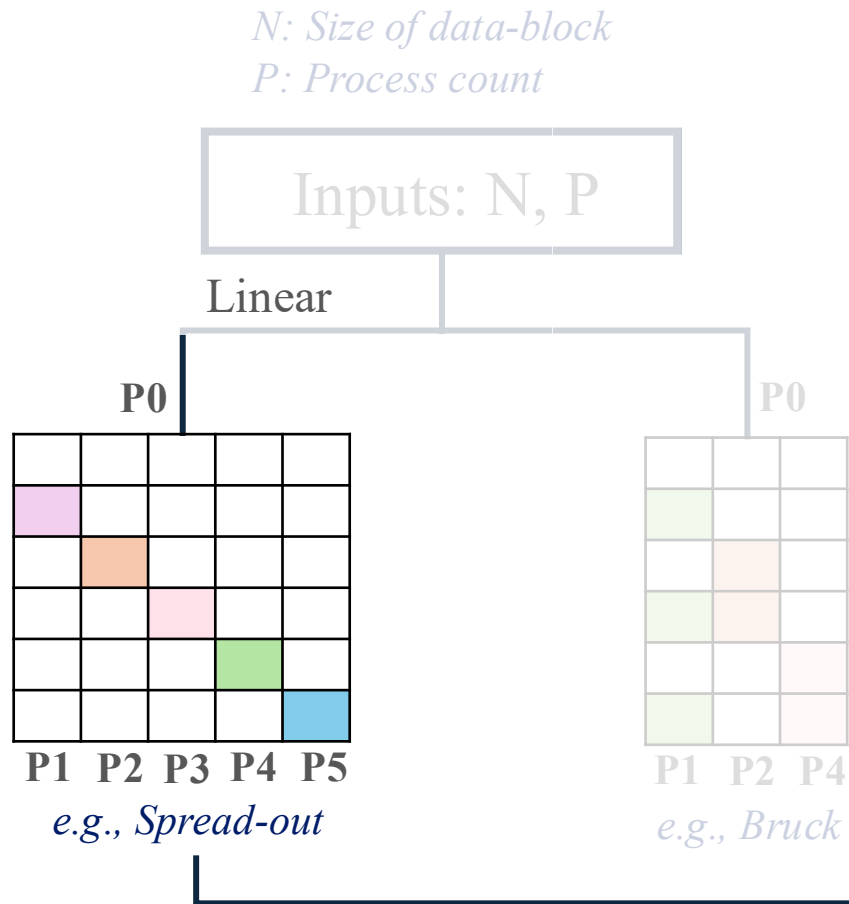
Spread-out algorithm



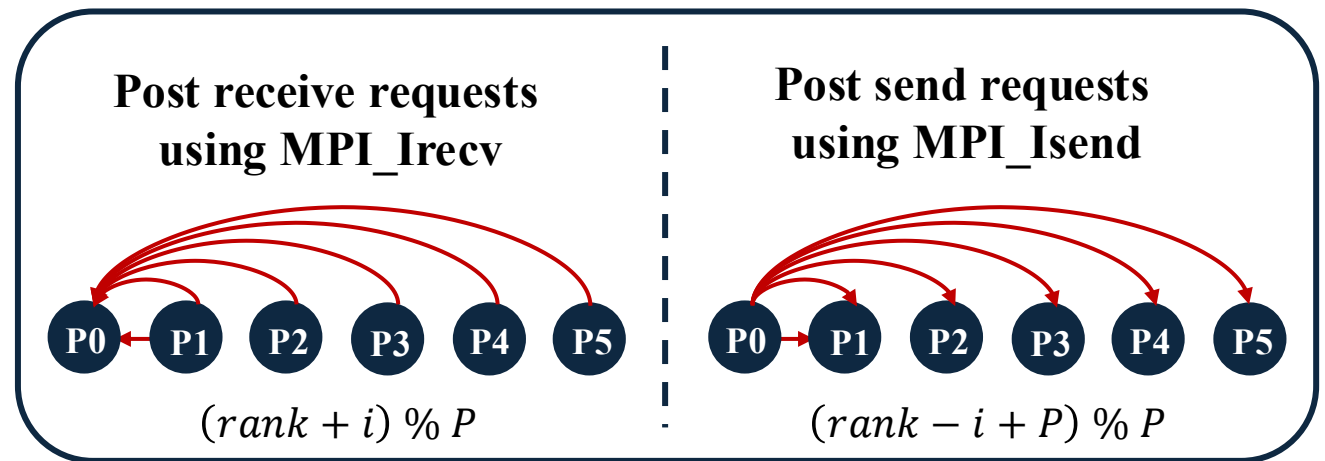
Standard MPI All-to-all Implementations: Spread-out



Standard MPI All-to-all Implementations: Spread-out

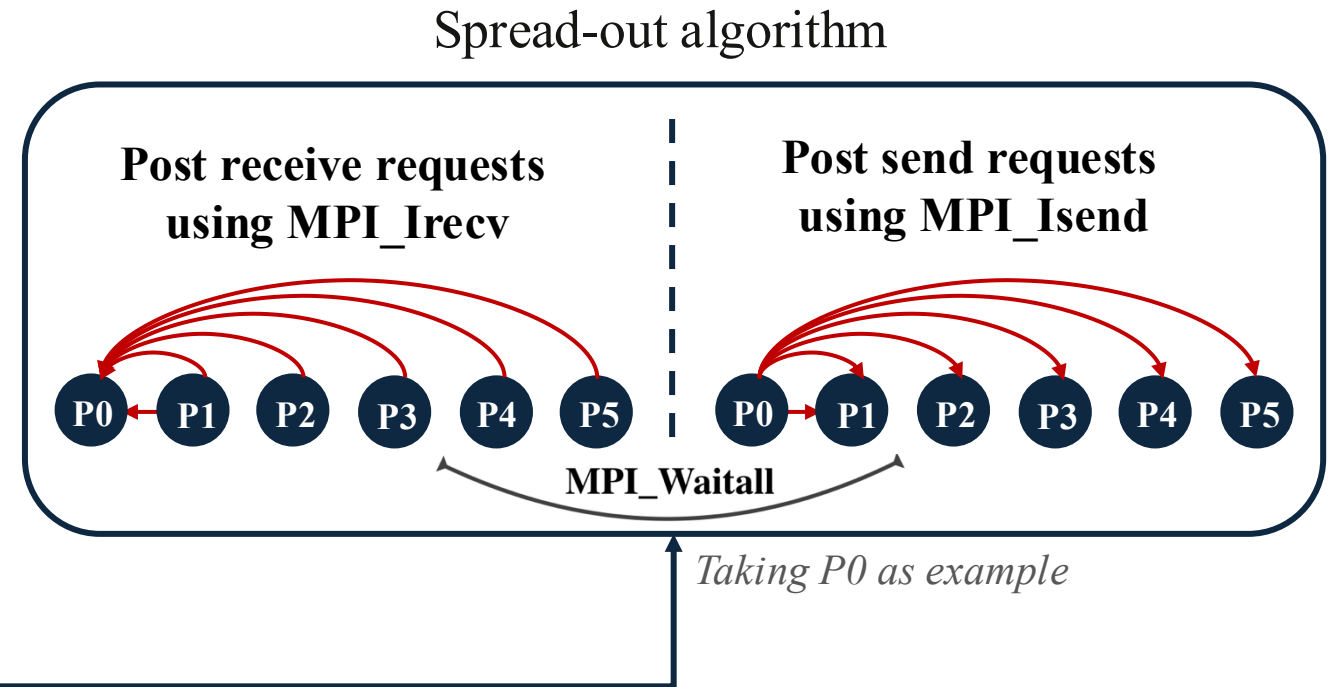
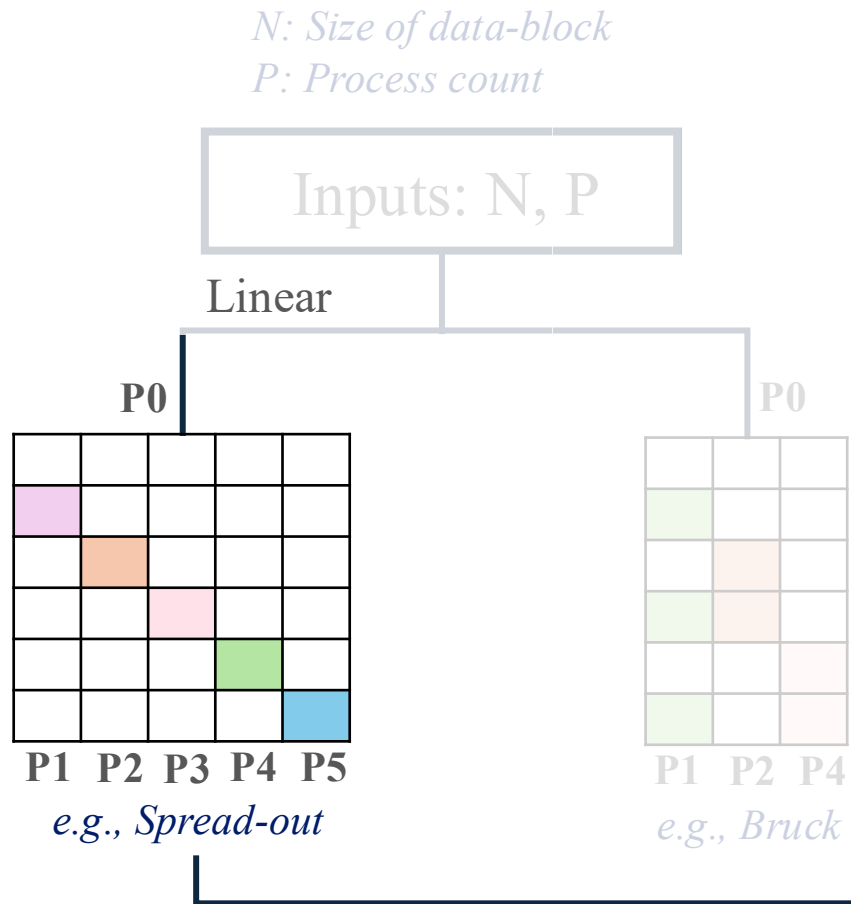


Spread-out algorithm

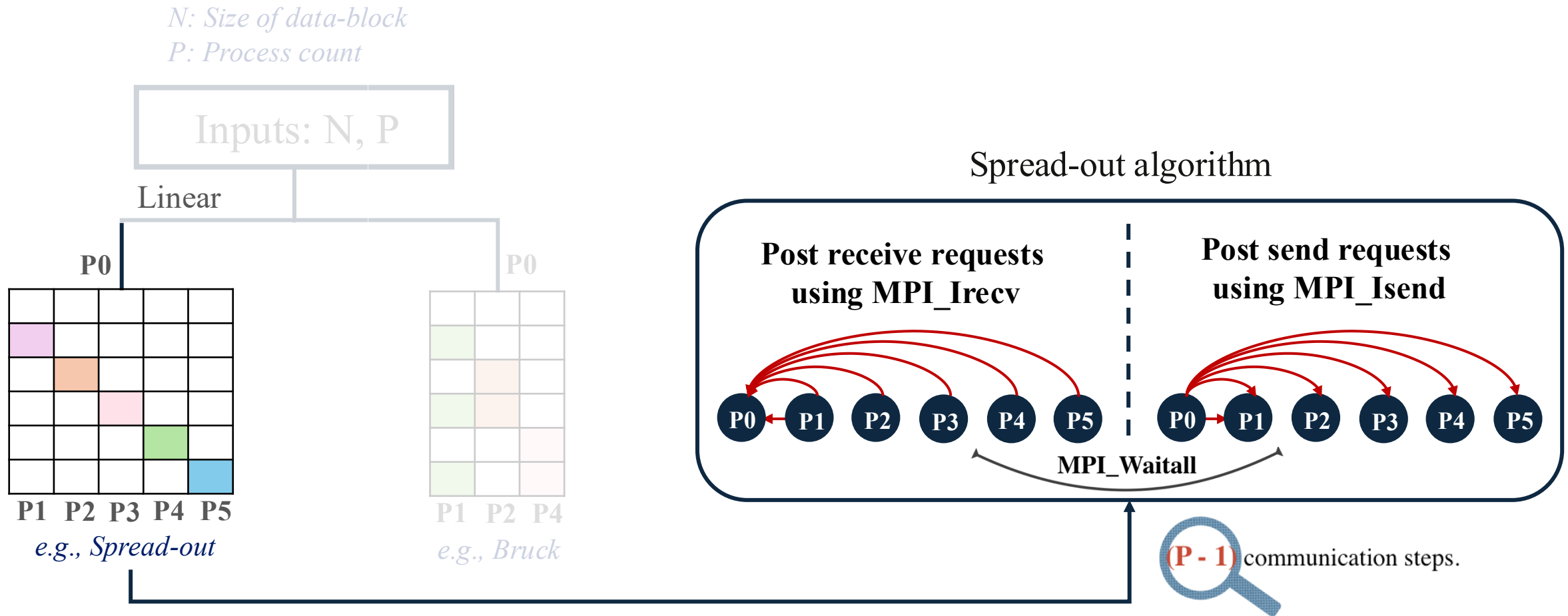


Taking P0 as example

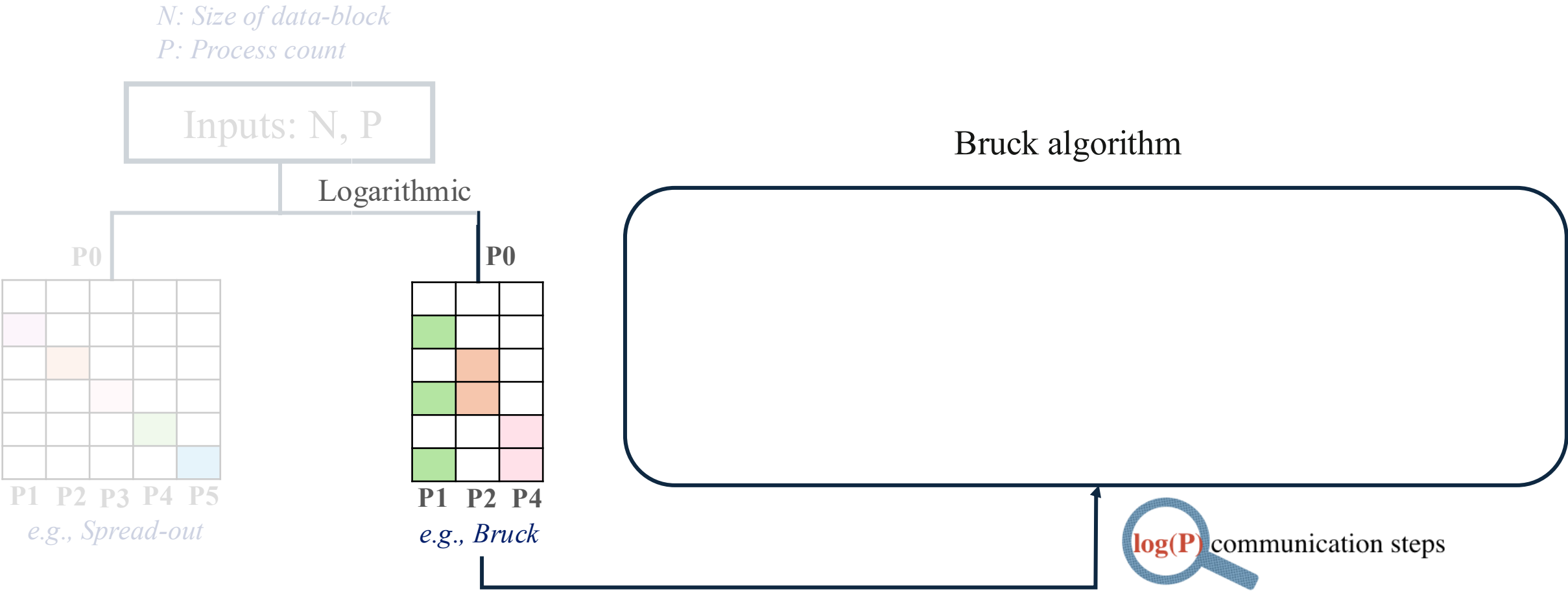
Standard MPI All-to-all Implementations: Spread-out



Standard MPI All-to-all Implementations: Spread-out



Standard MPI All-to-all Implementations: Bruck



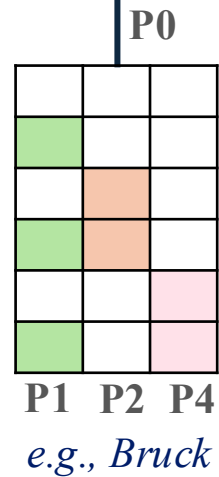
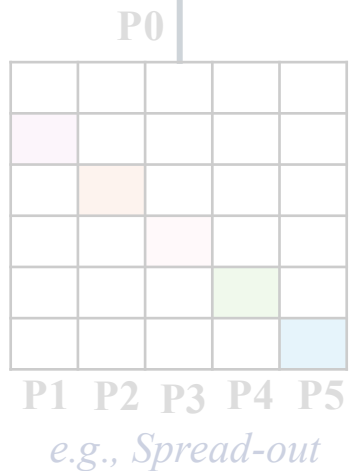
Standard MPI All-to-all Implementations: Bruck

N: Size of data-block

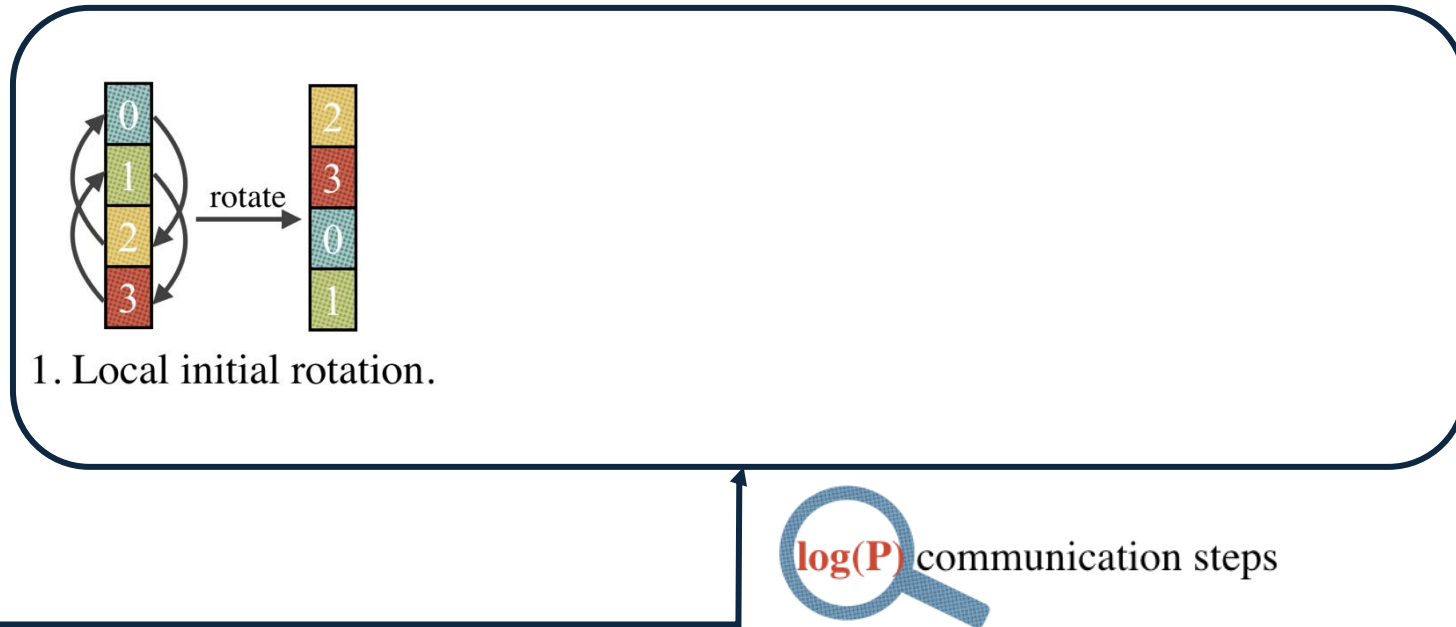
P: Process count

Inputs: N, P

Logarithmic



Bruck algorithm



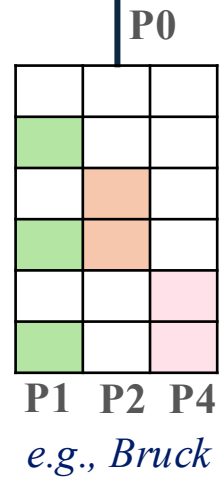
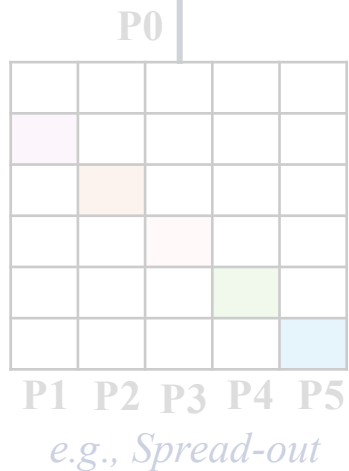
Standard MPI All-to-all Implementations: Bruck

N: Size of data-block

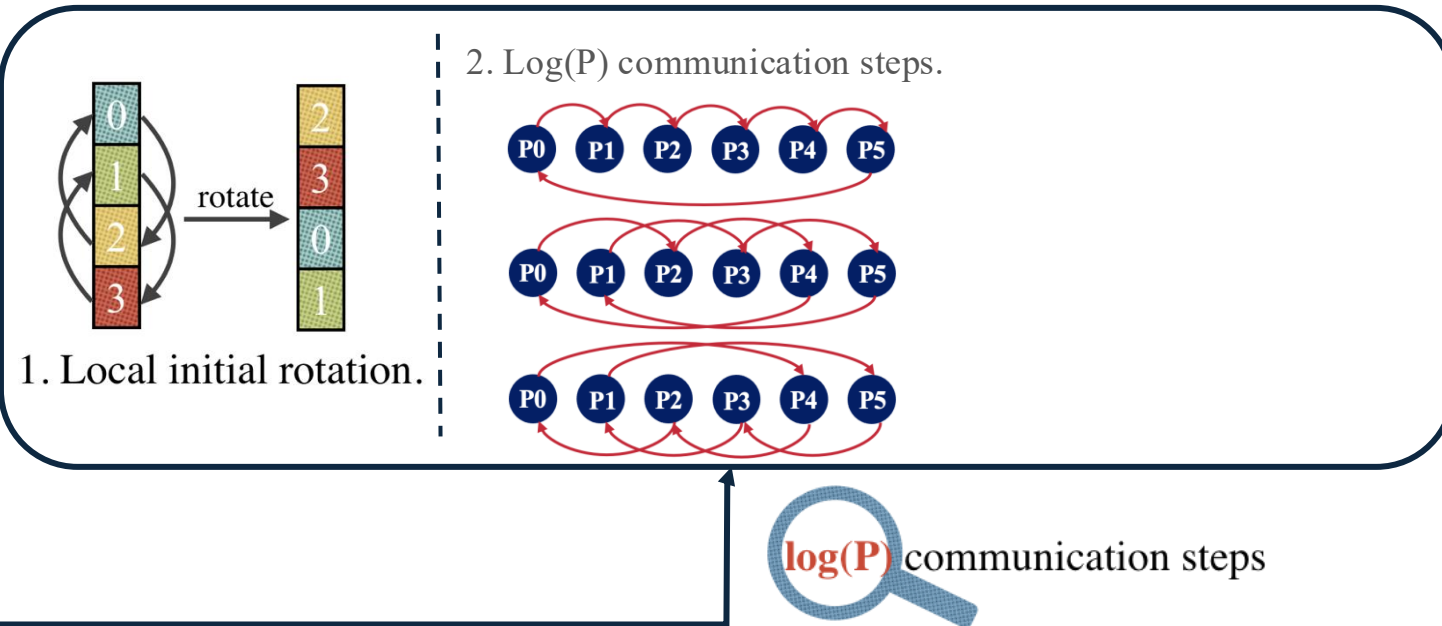
P: Process count

Inputs: N, P

Logarithmic



Bruck algorithm



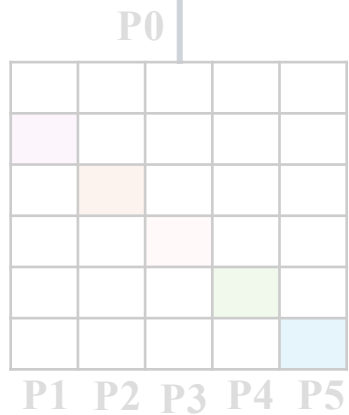
Standard MPI All-to-all Implementations: Comm-0

N: Size of data-block

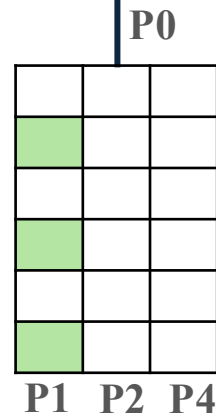
P: Process count

Inputs: N, P

Logarithmic

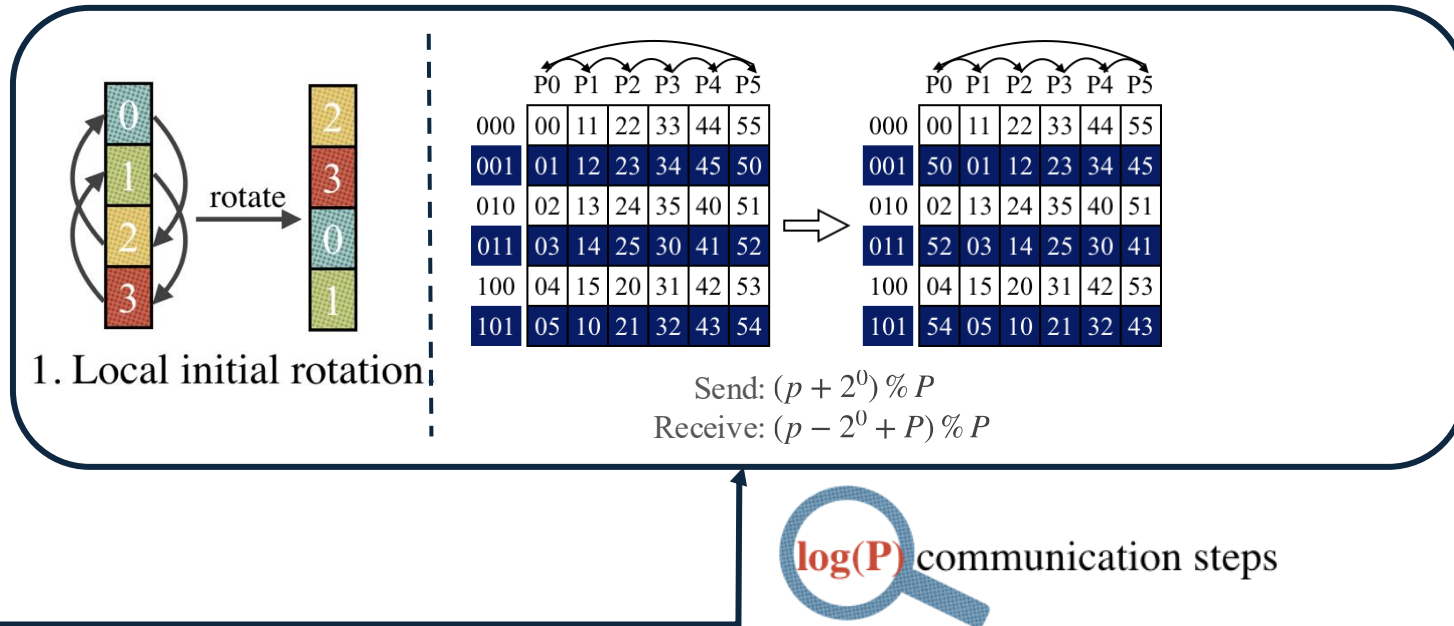


e.g., Spread-out



e.g., Bruck

Bruck algorithm



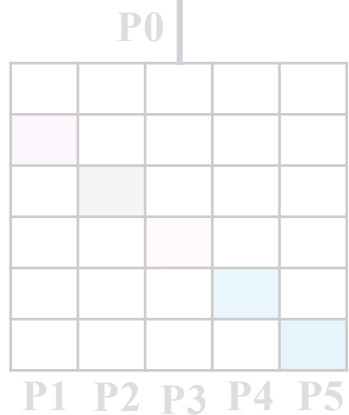
Standard MPI All-to-all Implementations: Comm-1

N: Size of data-block

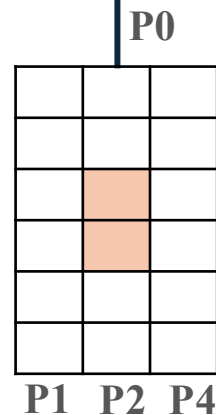
P: Process count

Inputs: N, P

Logarithmic

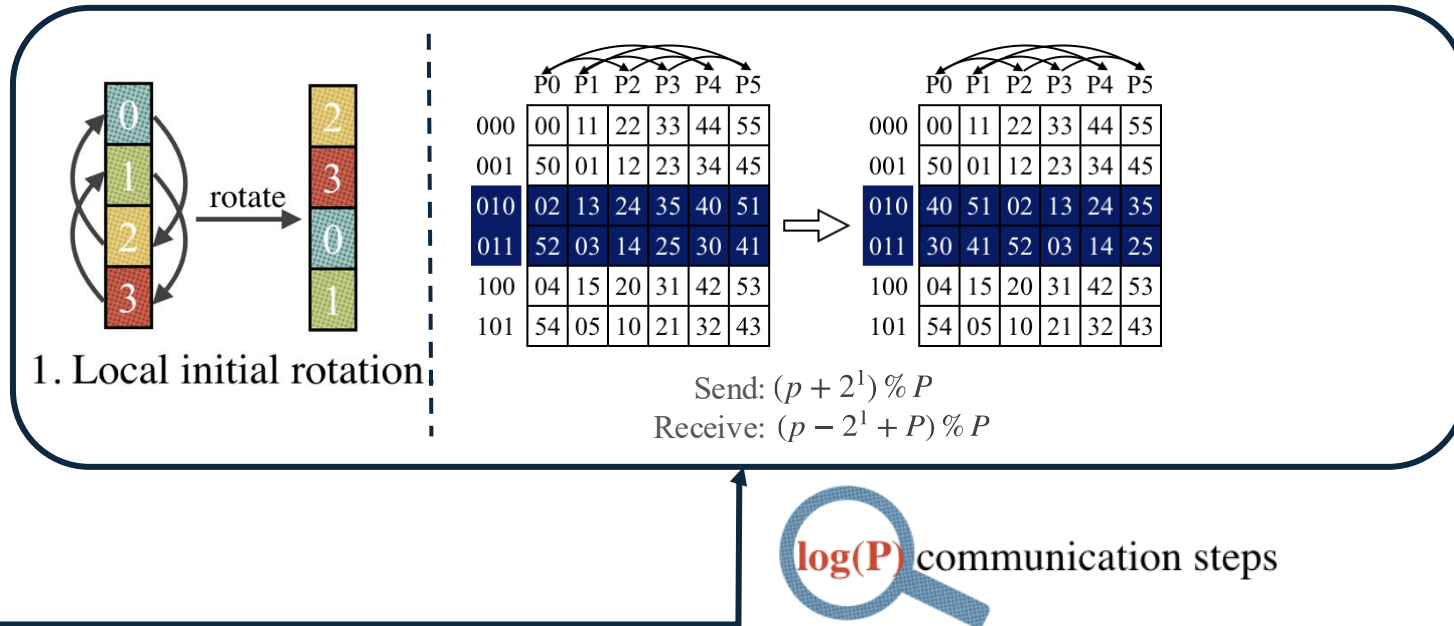


e.g., Spread-out



e.g., Bruck

Bruck algorithm



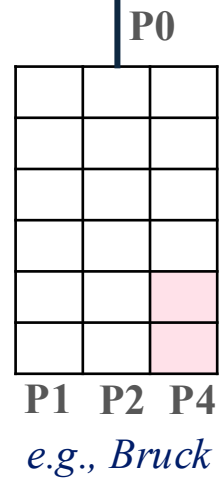
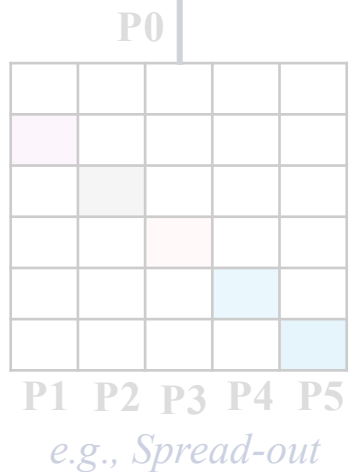
Standard MPI All-to-all Implementations: Comm-2

N: Size of data-block

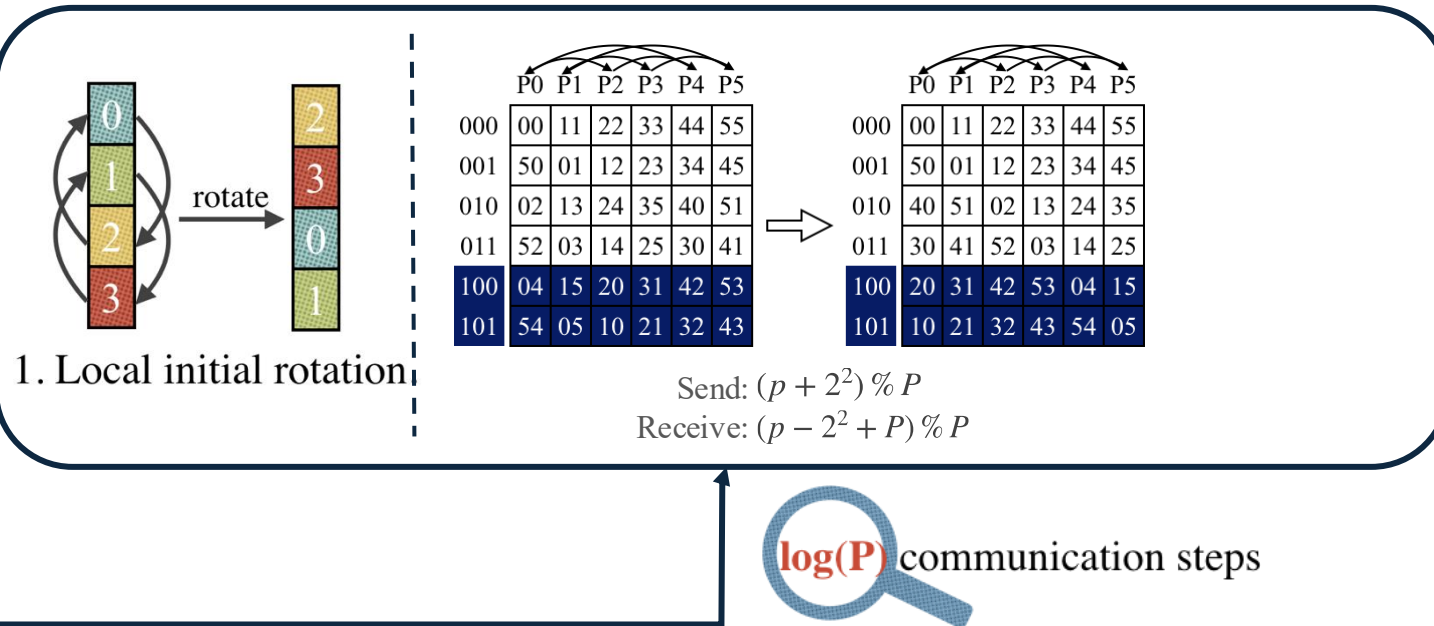
P: Process count

Inputs: N, P

Logarithmic



Bruck algorithm



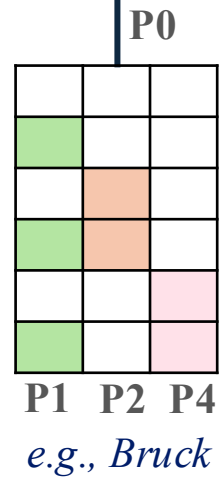
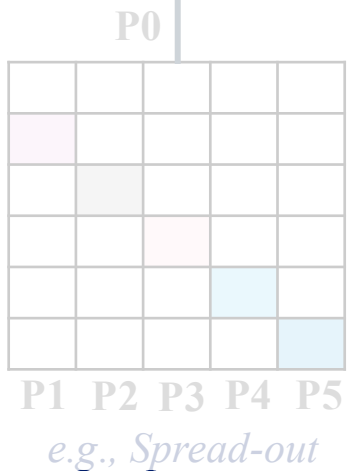
Standard MPI All-to-all Implementations: Bruck

N: Size of data-block

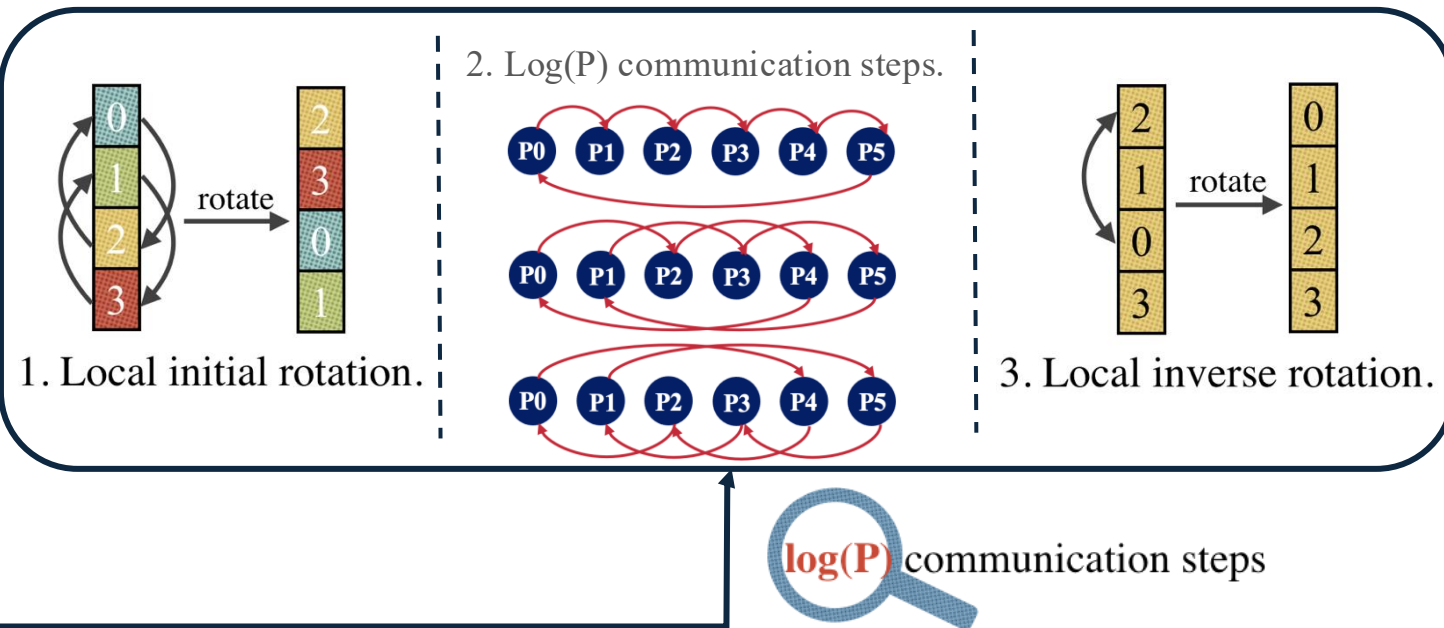
P: Process count

Inputs: N, P

Logarithmic



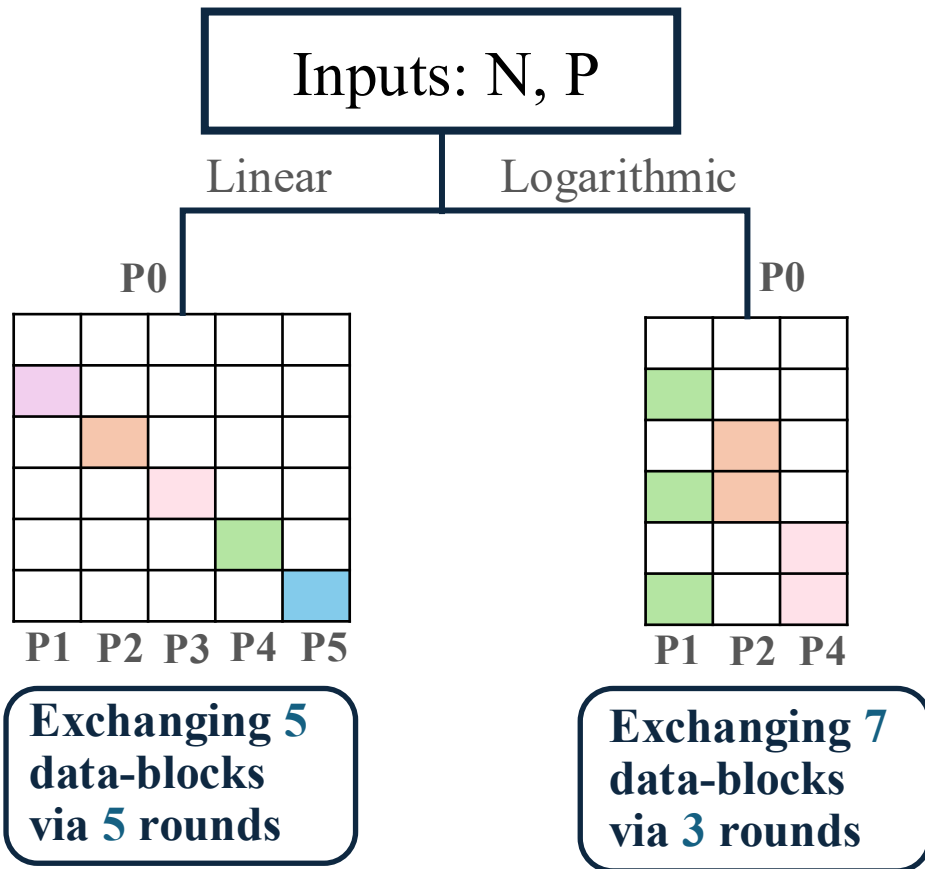
Bruck algorithm



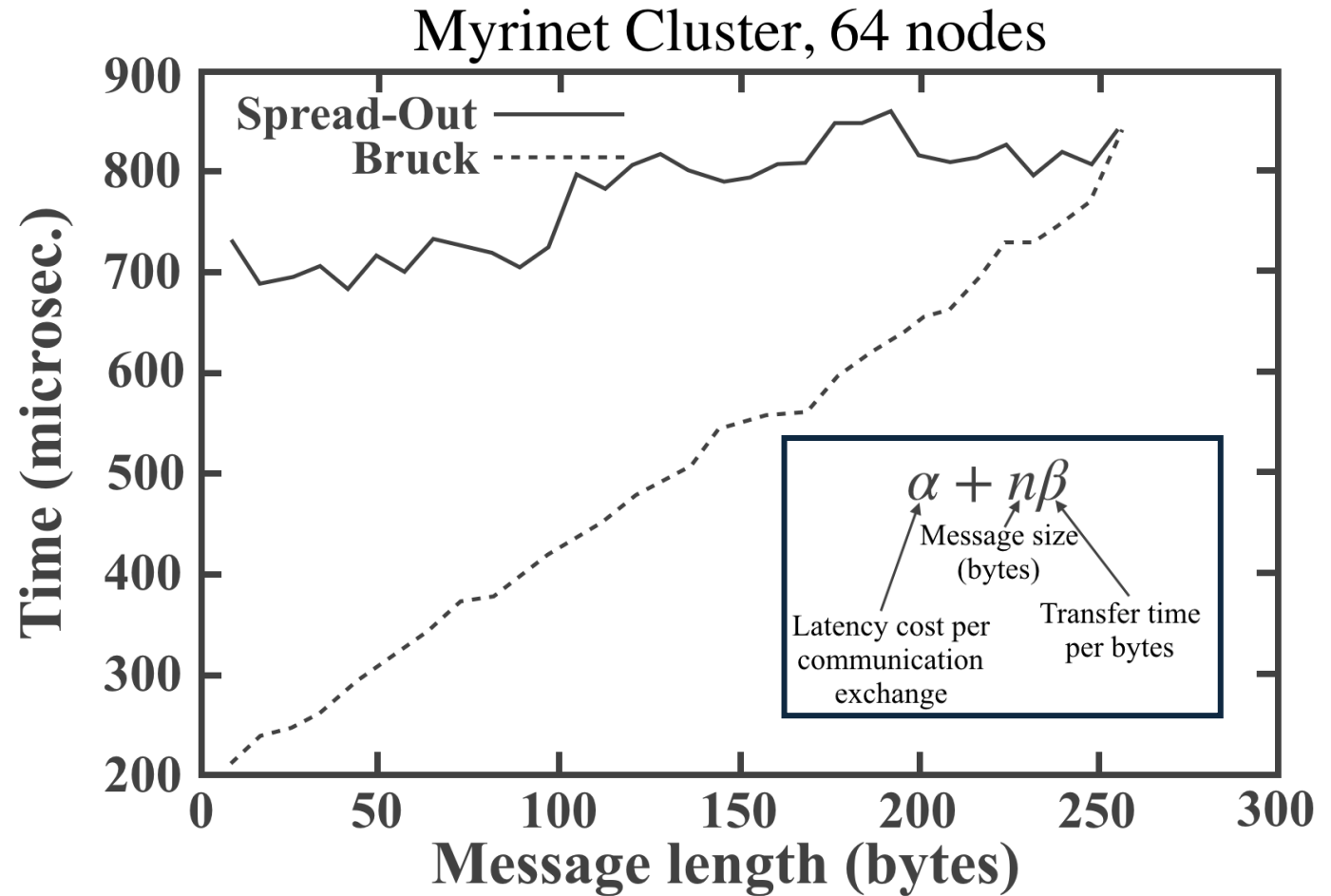
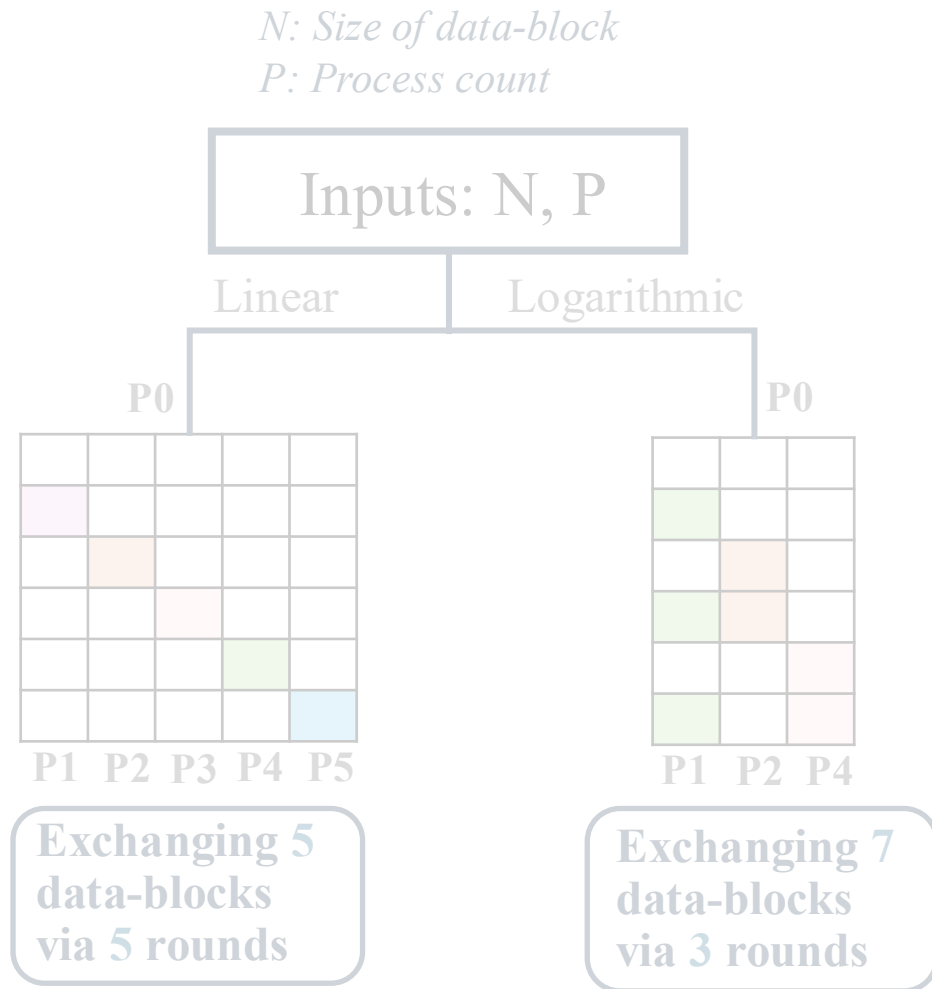
Comparison of These Two Algorithms

N: Size of data-block

P: Process count



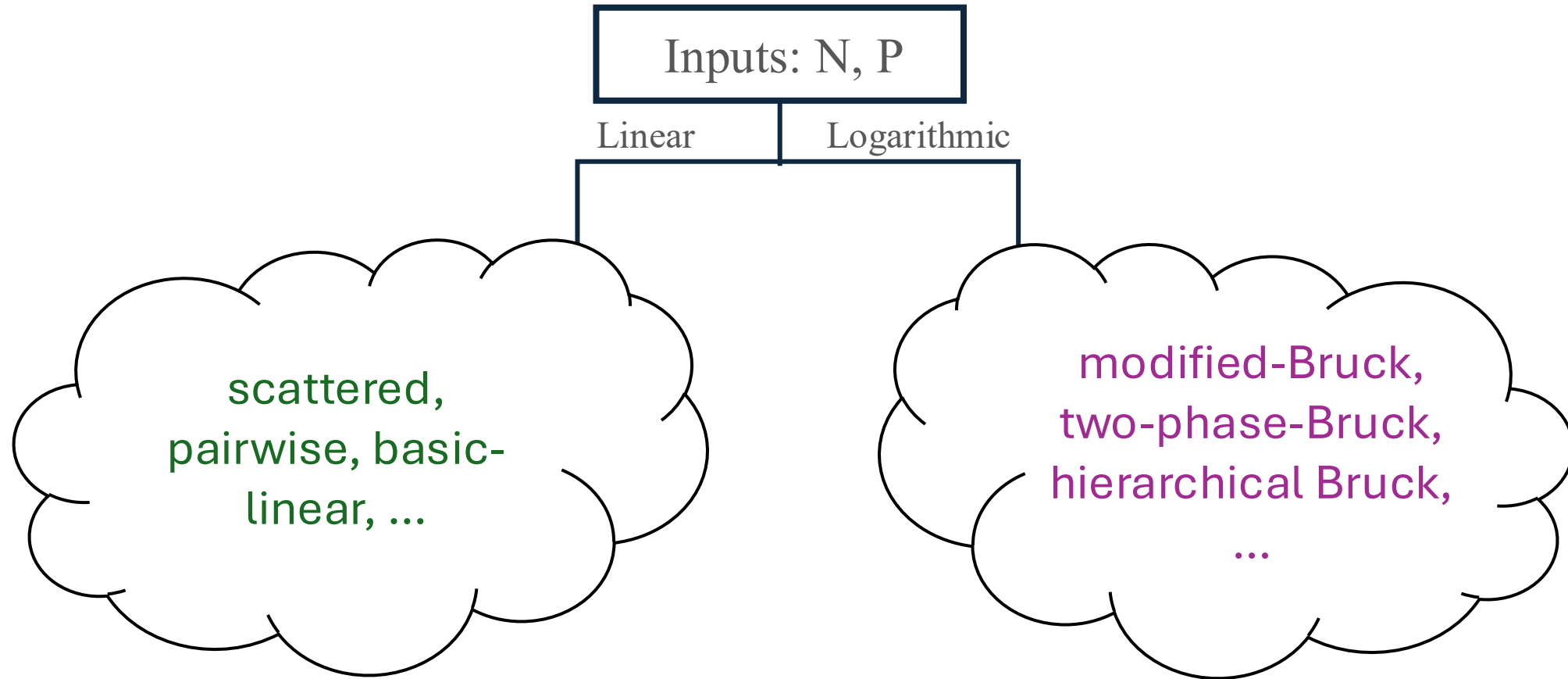
Comparison of These Two Algorithms



Many All-to-all Implementations

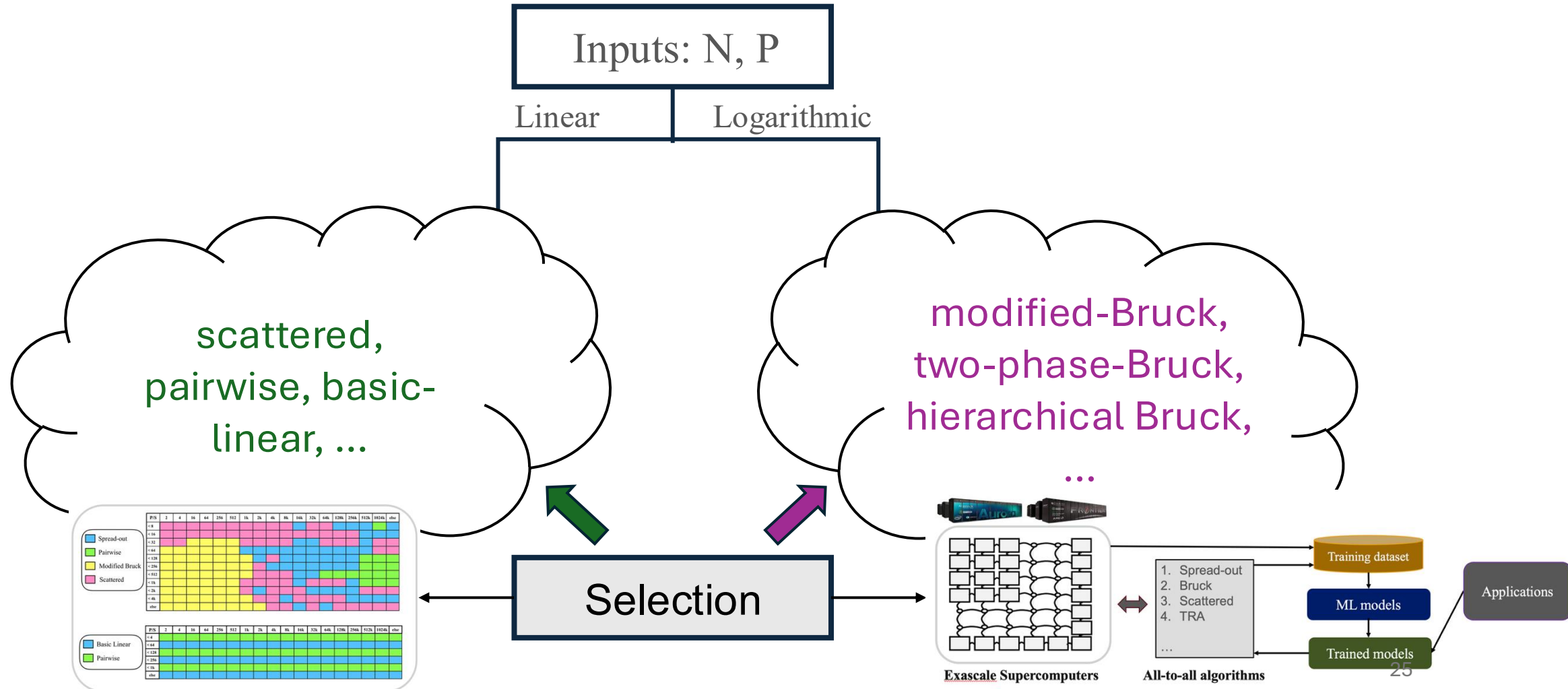
N: Size of data-block

P: Process count

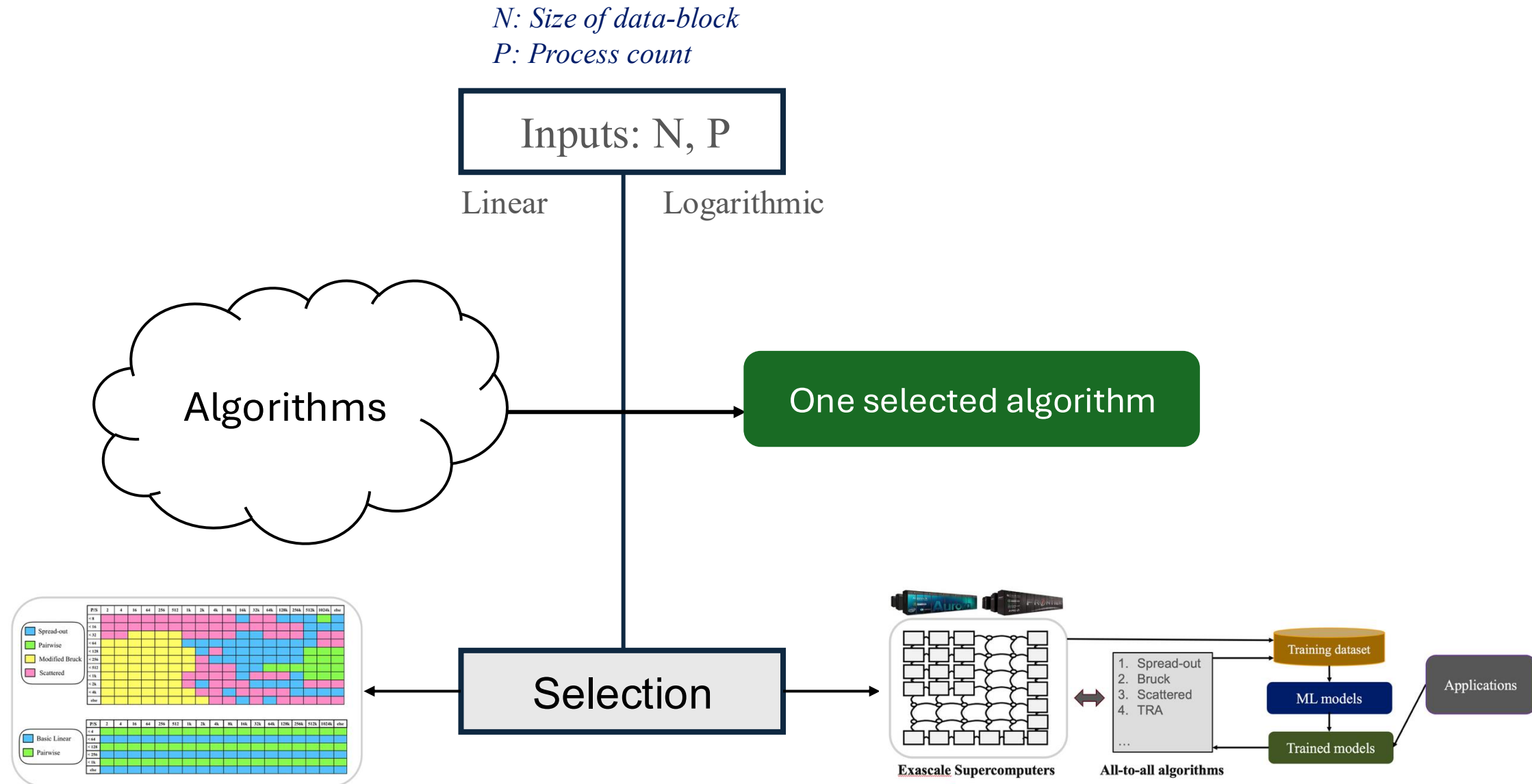


All-to-all Implementation: Algorithm Selection

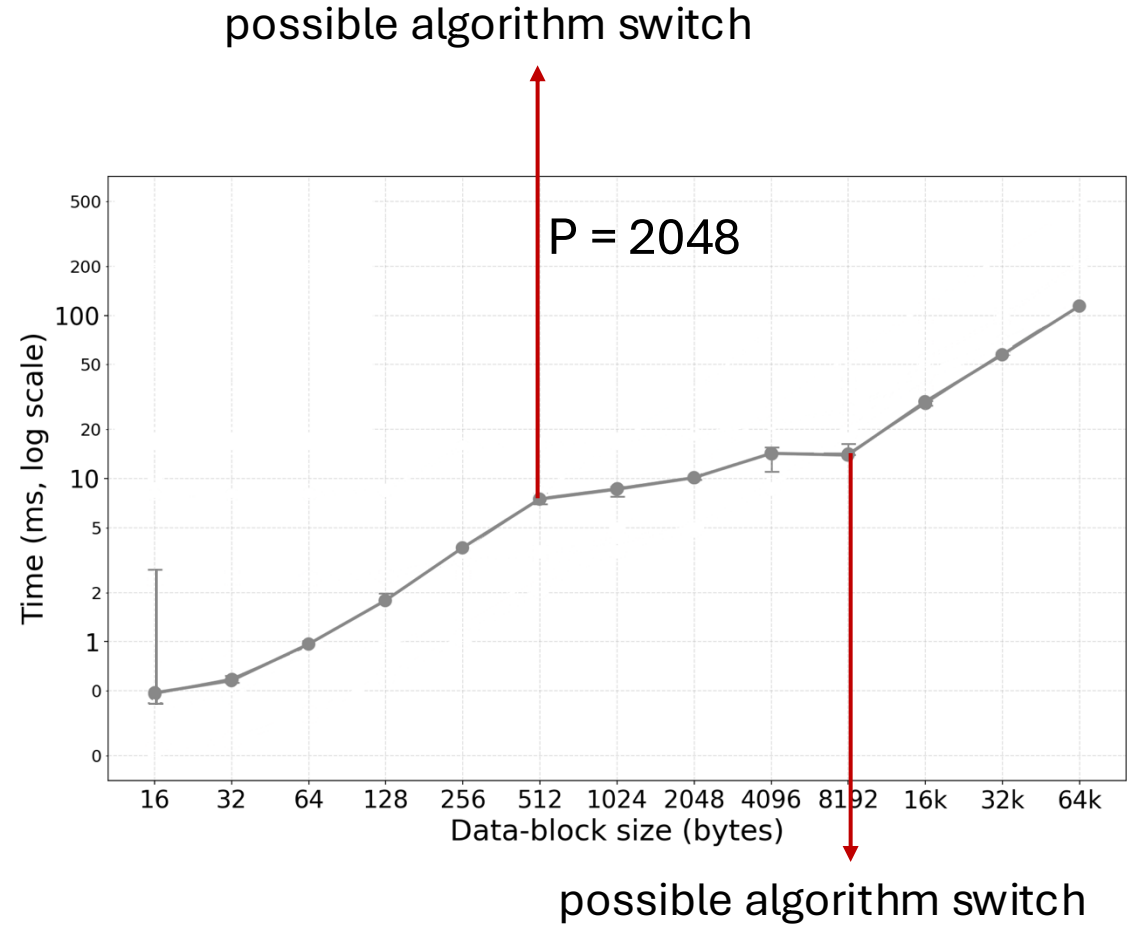
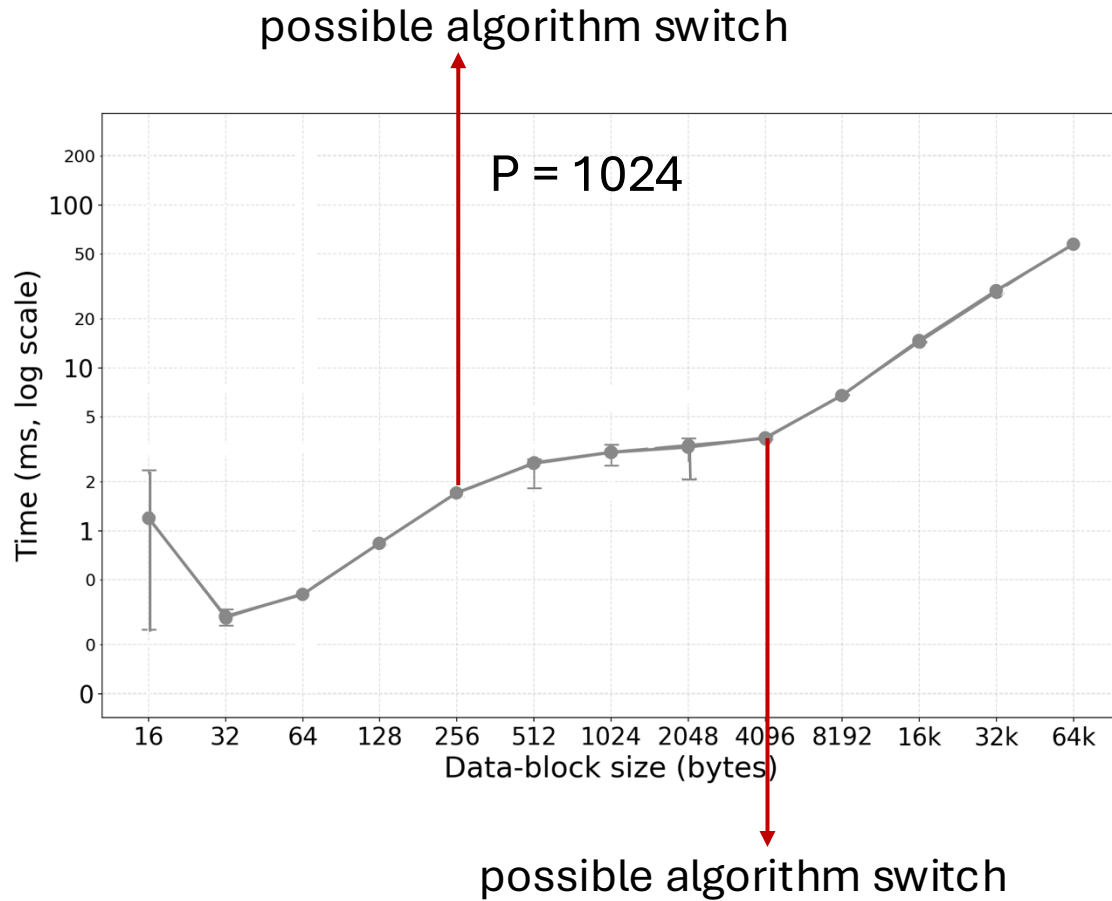
N : Size of data-block
 P : Process count



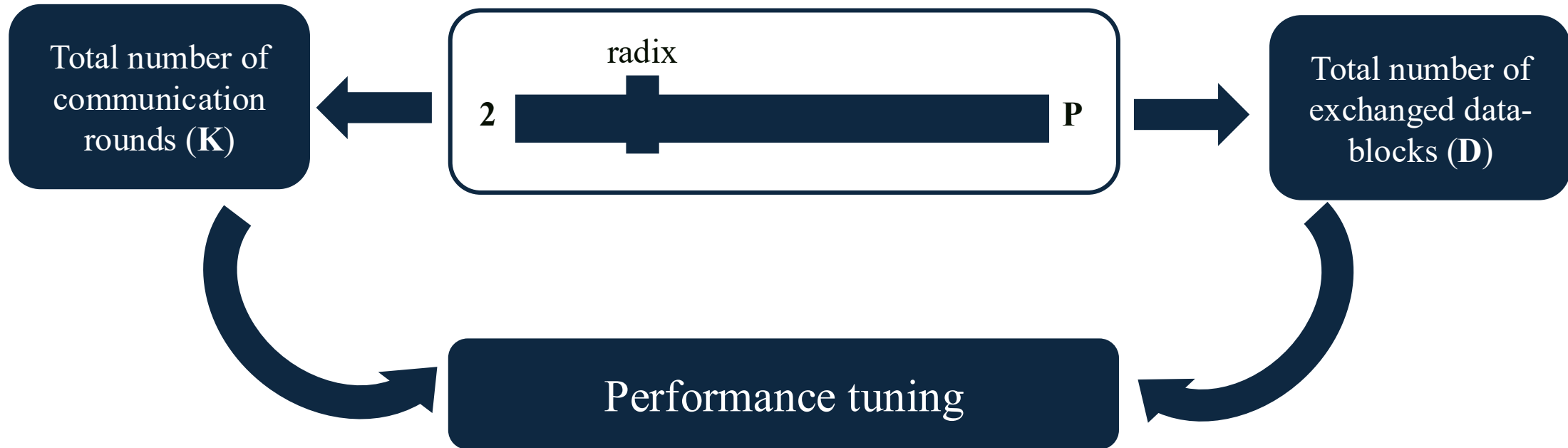
All-to-all Implementation: Algorithm Selection



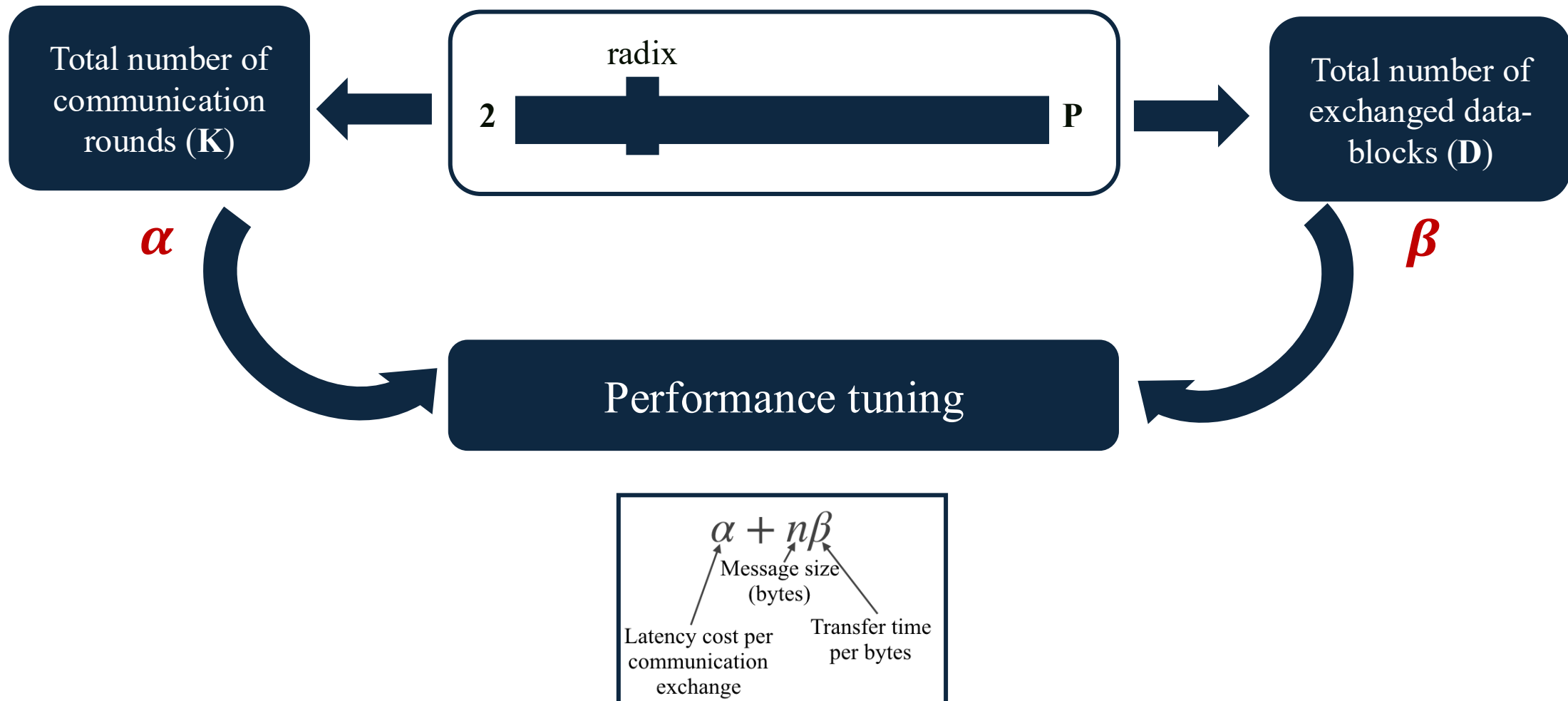
Algorithm Switching



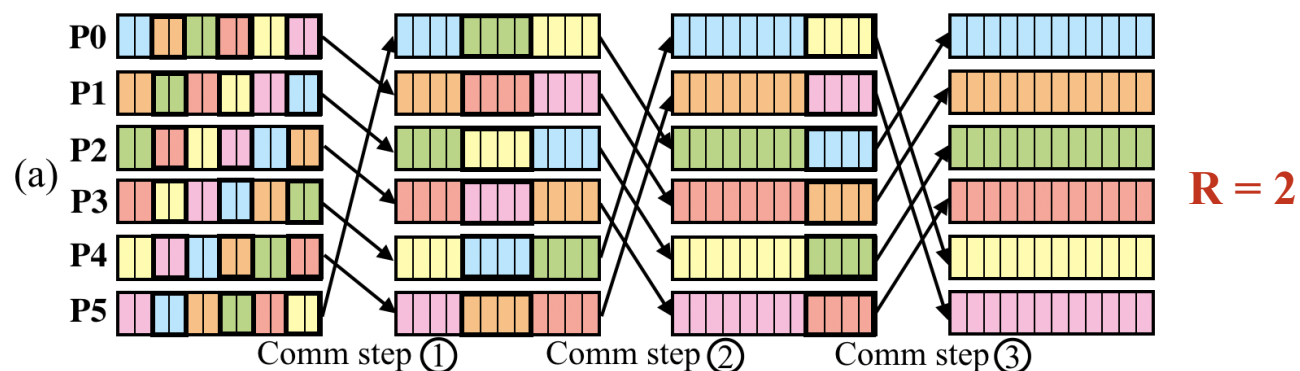
Prior Work Enables Tunable Radices For Bruck-style Algorithms



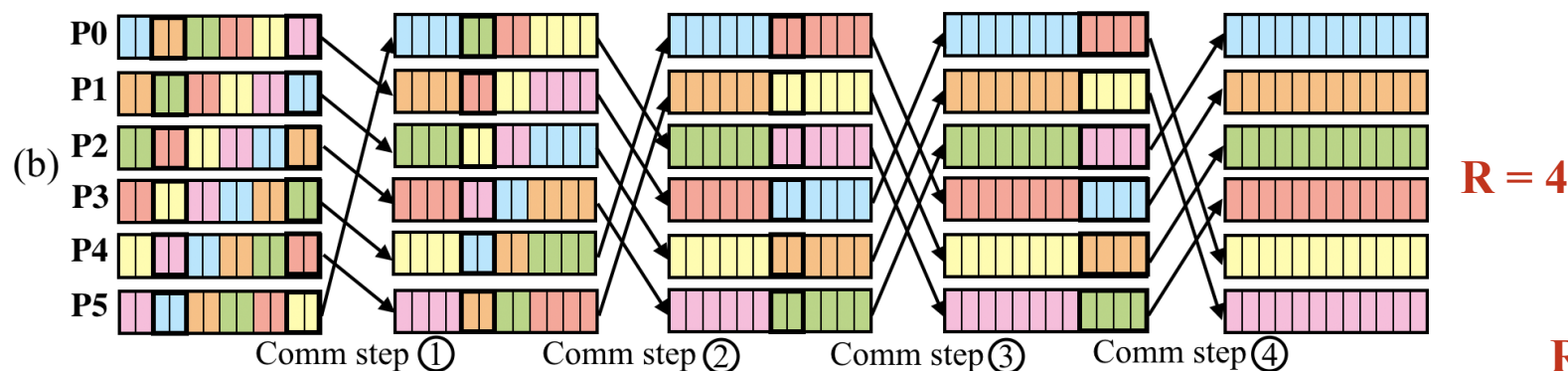
Prior Work Enables Tunable Radices For Bruck-style Algorithms



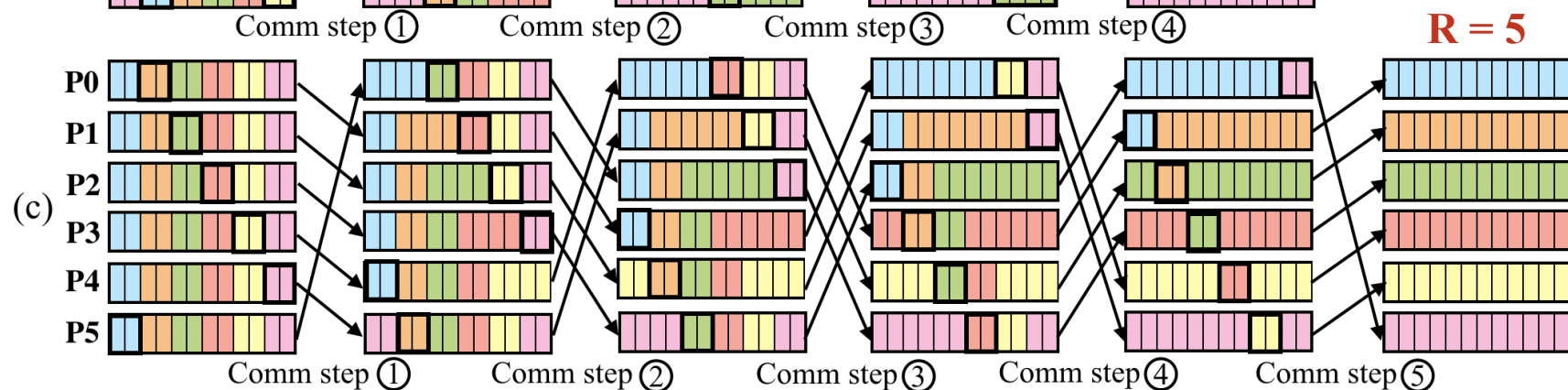
Examples of Tunable Radices



Exchanging 10
data-blocks via
3 rounds



Exchanging 6
data-blocks via
4 rounds

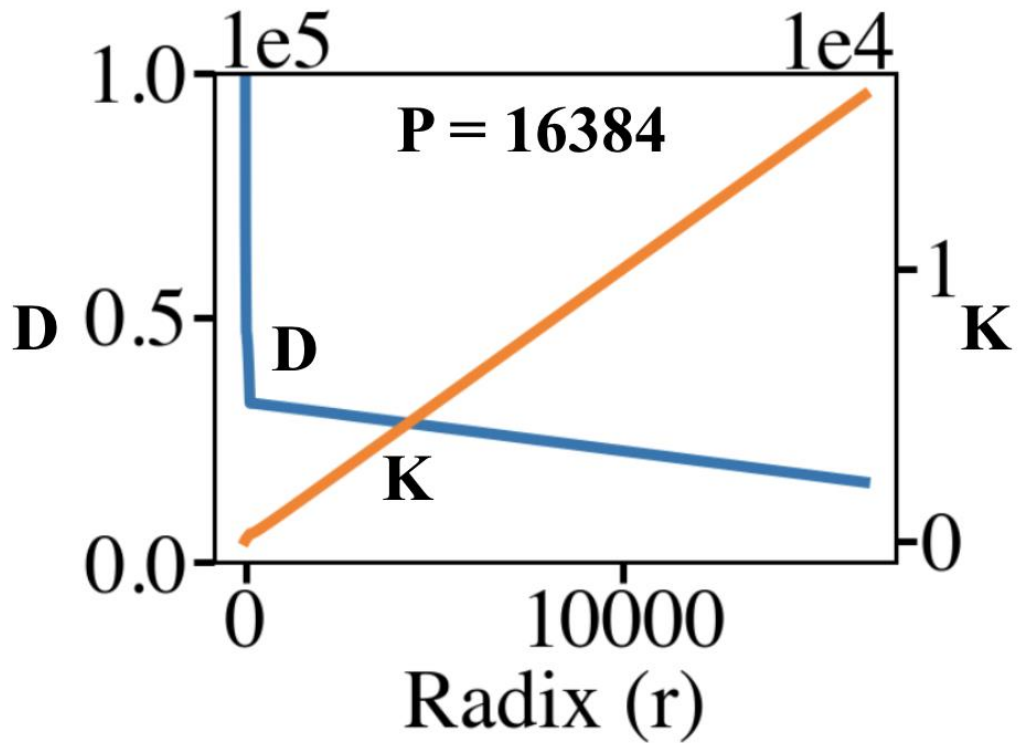


Exchanging 5
data-blocks via
5 rounds

Sqrt(P) Delivers Best Overall Performance

K – total number of communication rounds

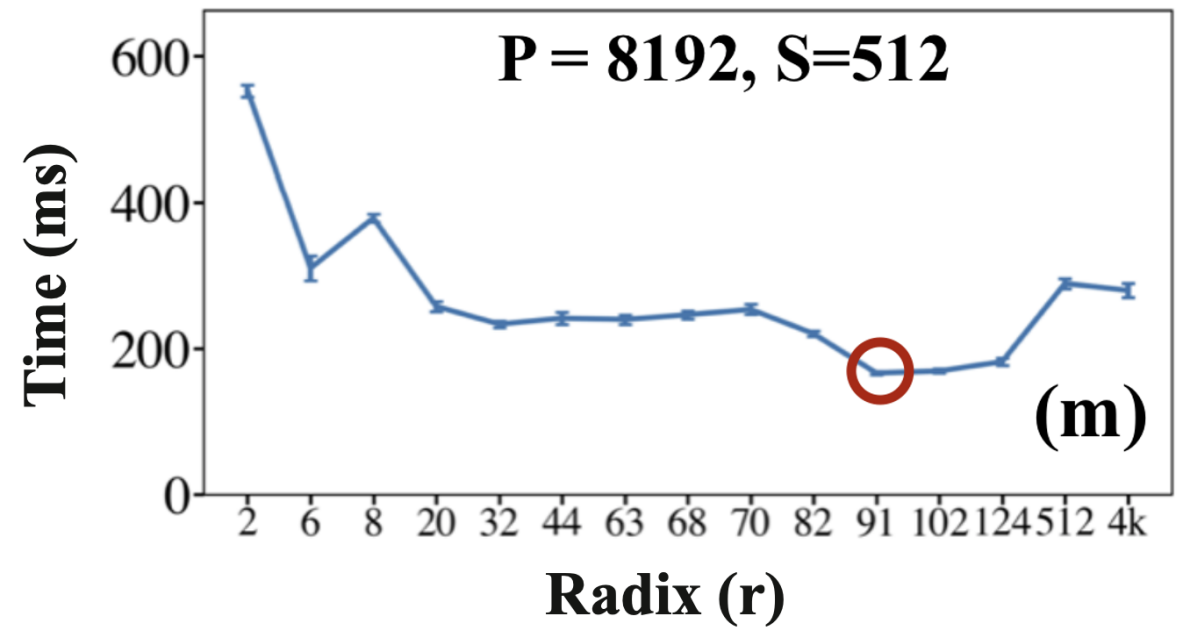
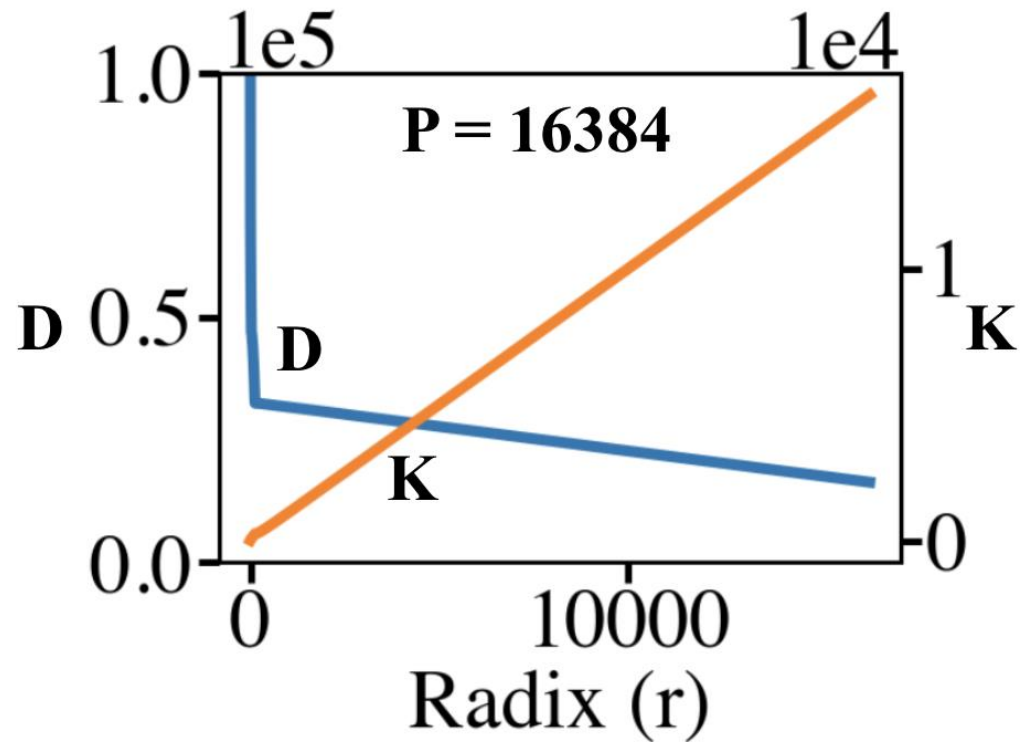
D – total amount of data transmitted



Sqrt(P) Delivers Best Overall Performance

K – total number of communication rounds

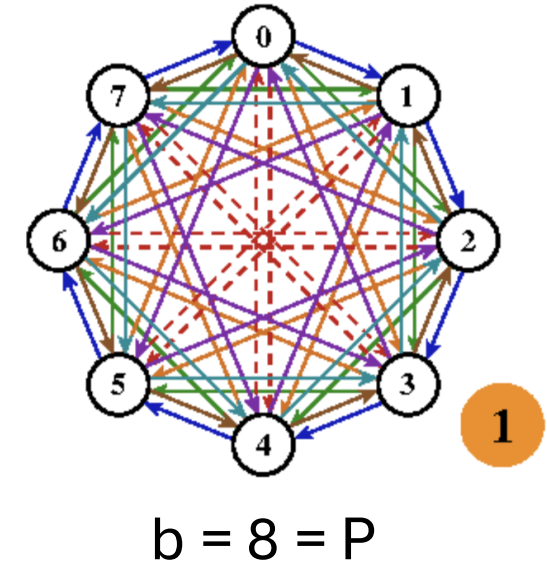
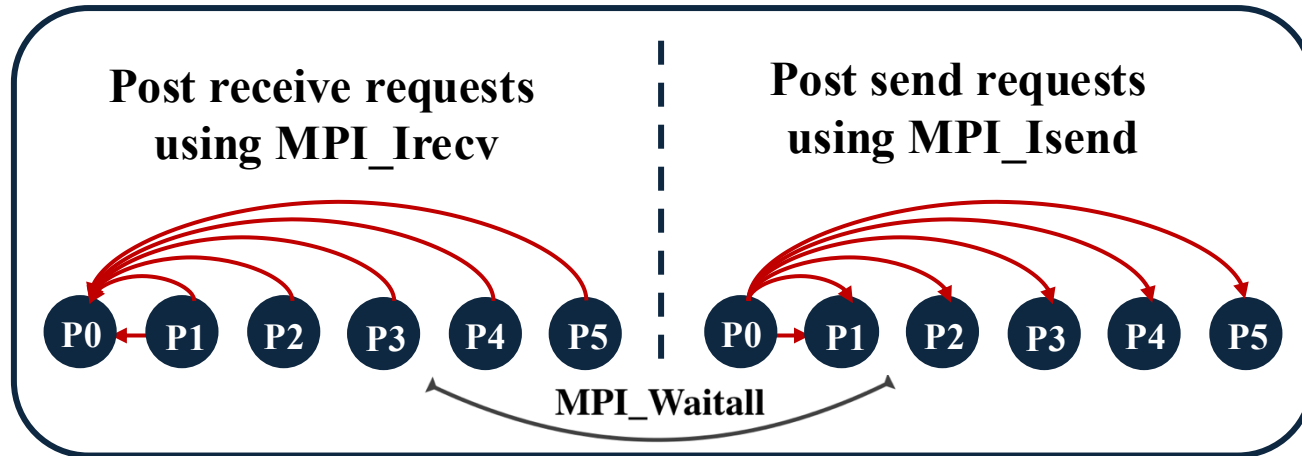
D – total amount of data transmitted



$r = \lceil \sqrt{8192} \rceil = 91$ matches the theoretical deduction

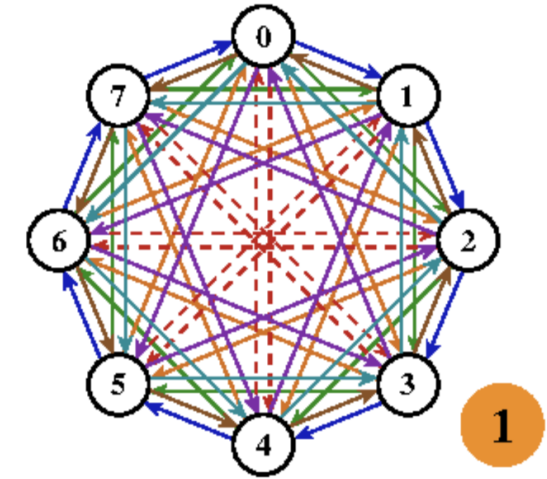
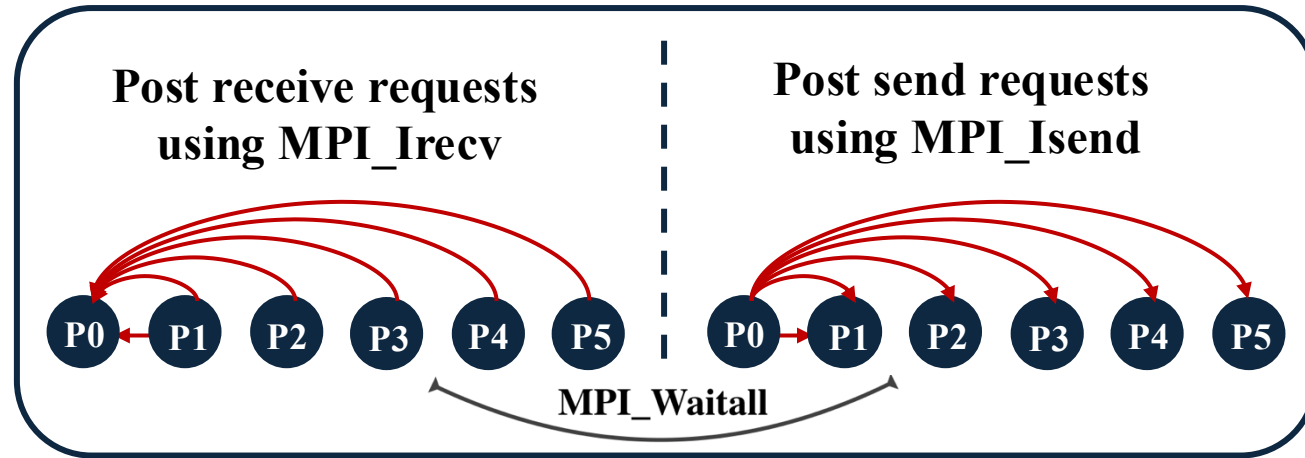
Linear Implementation Enables Tunable Batch-size

Spread-out algorithm



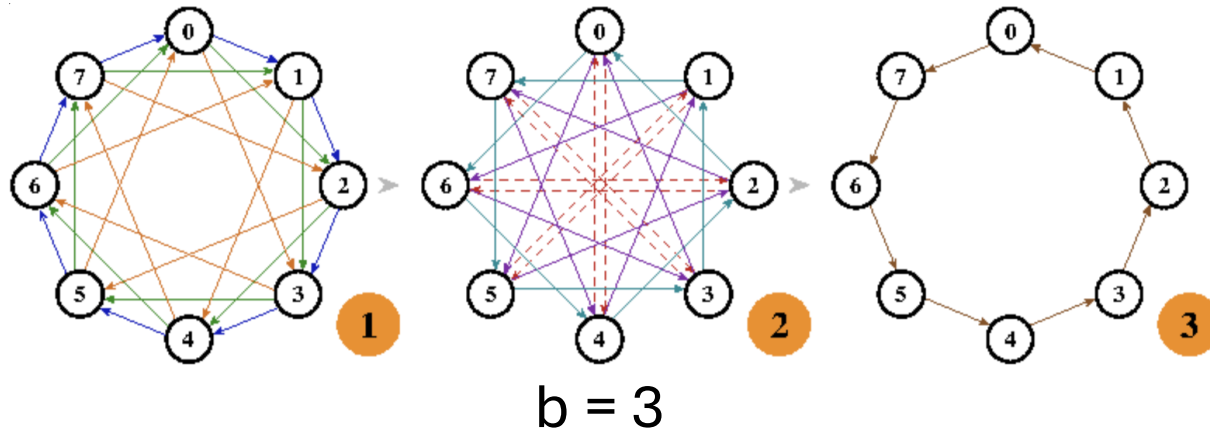
Linear Implementation Enables Tunable Batch-size

Spread-out algorithm



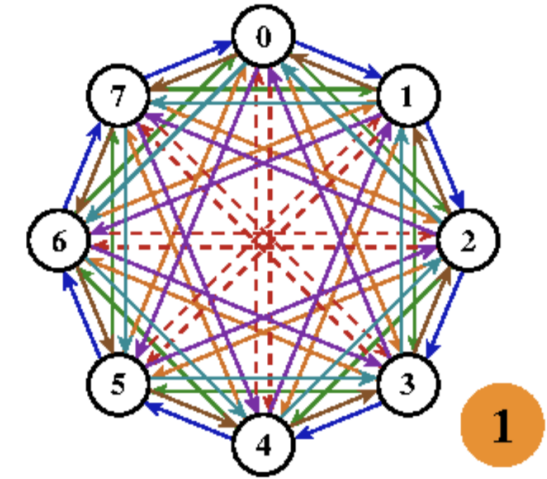
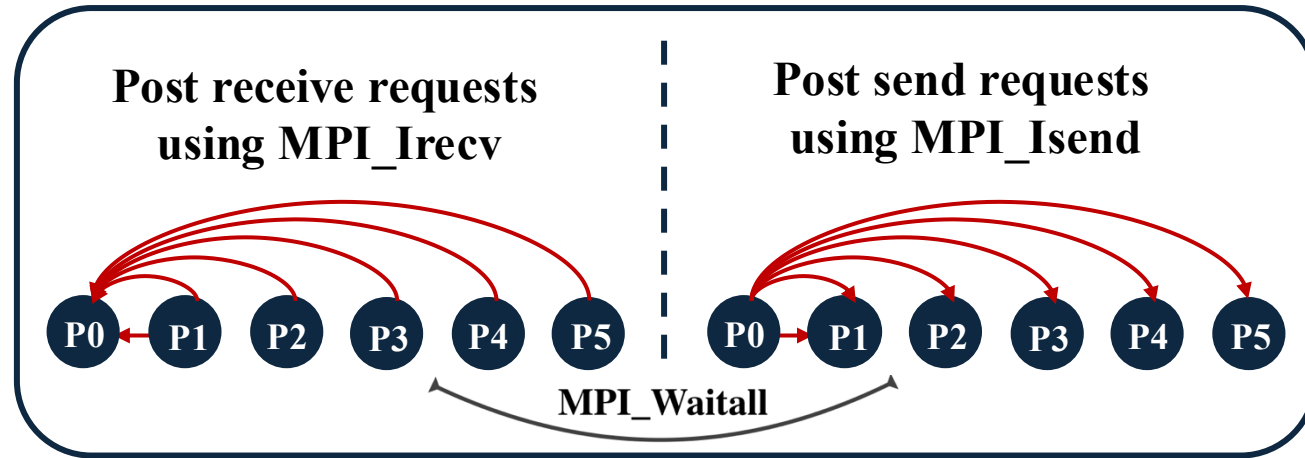
$$b = 8 = P$$

Scattered algorithm

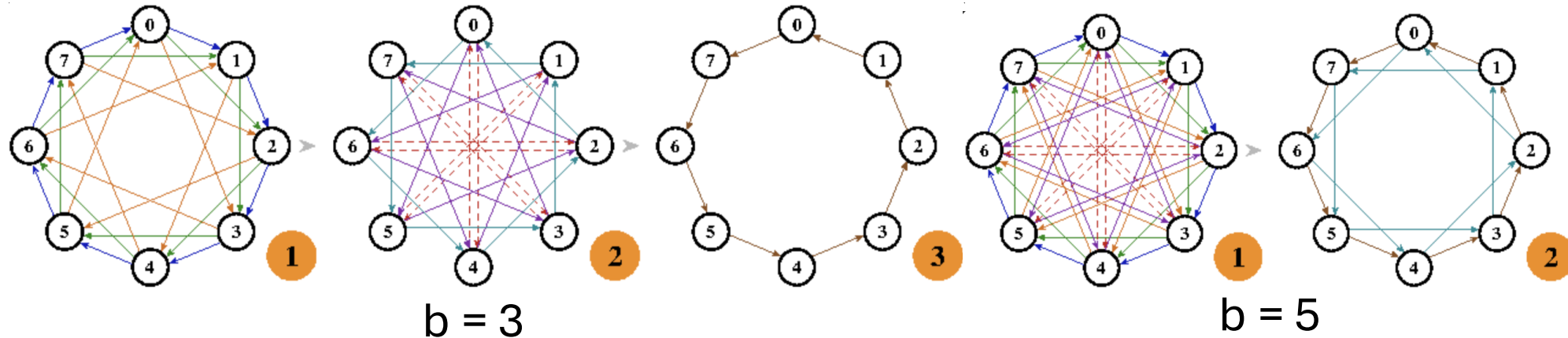
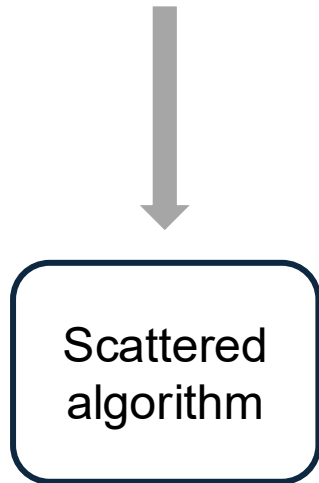


Linear Implementation Enables Tunable Batch-size

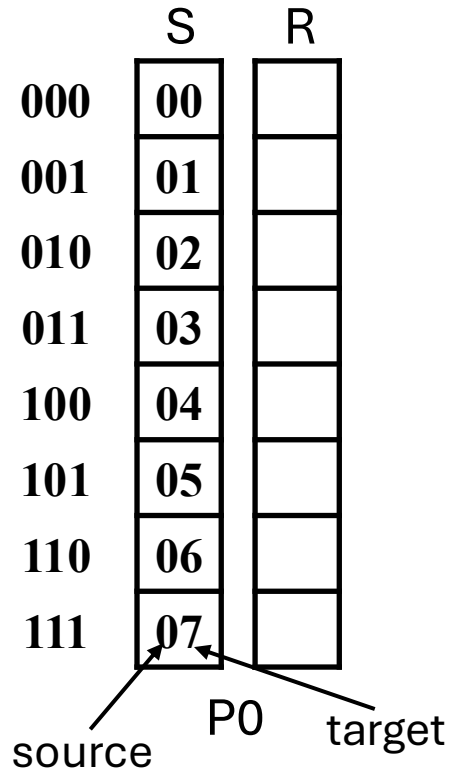
Spread-out algorithm



$b = 8 = P$

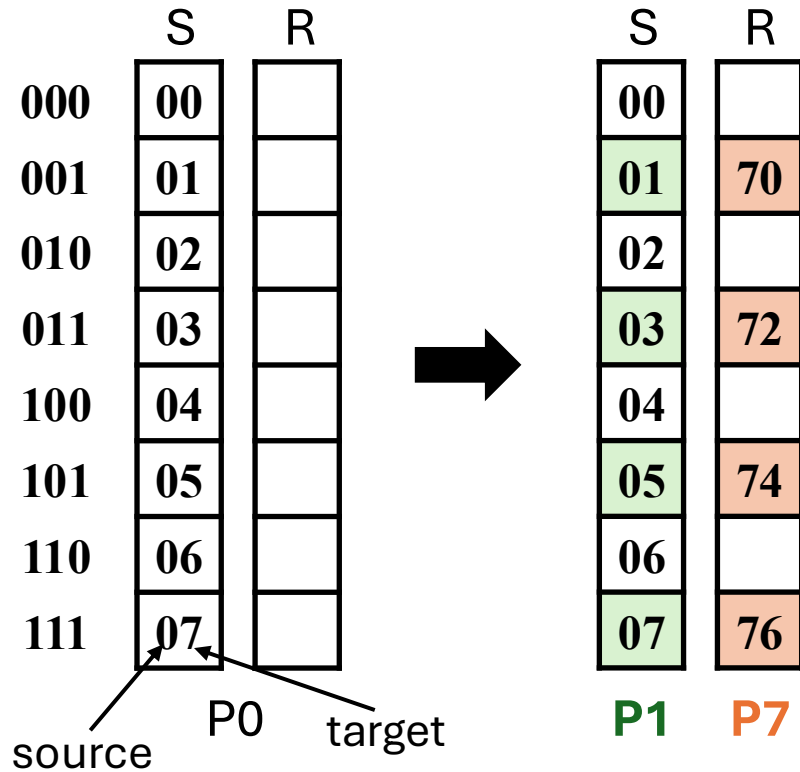


Bruck-style Algorithm Implies Blocking Communication



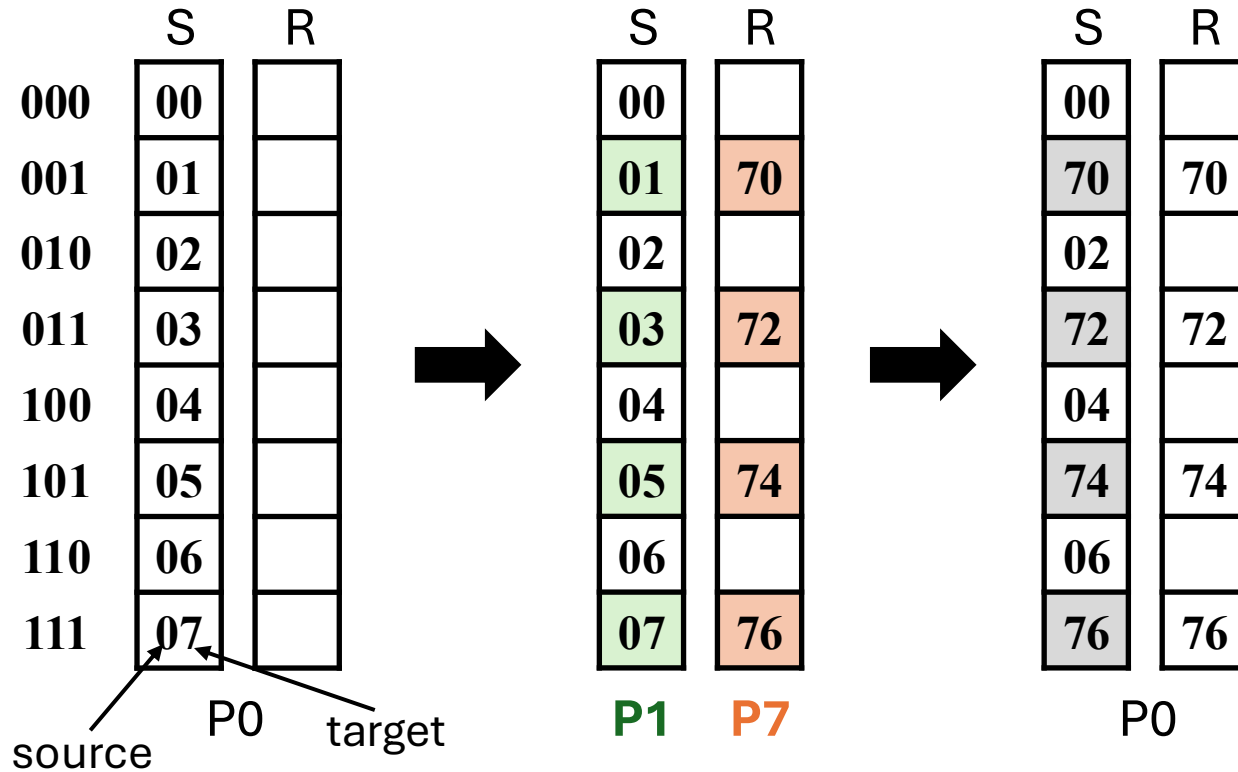
$$P = 8, r = 2$$

Bruck-style Algorithm Implies Blocking Communication



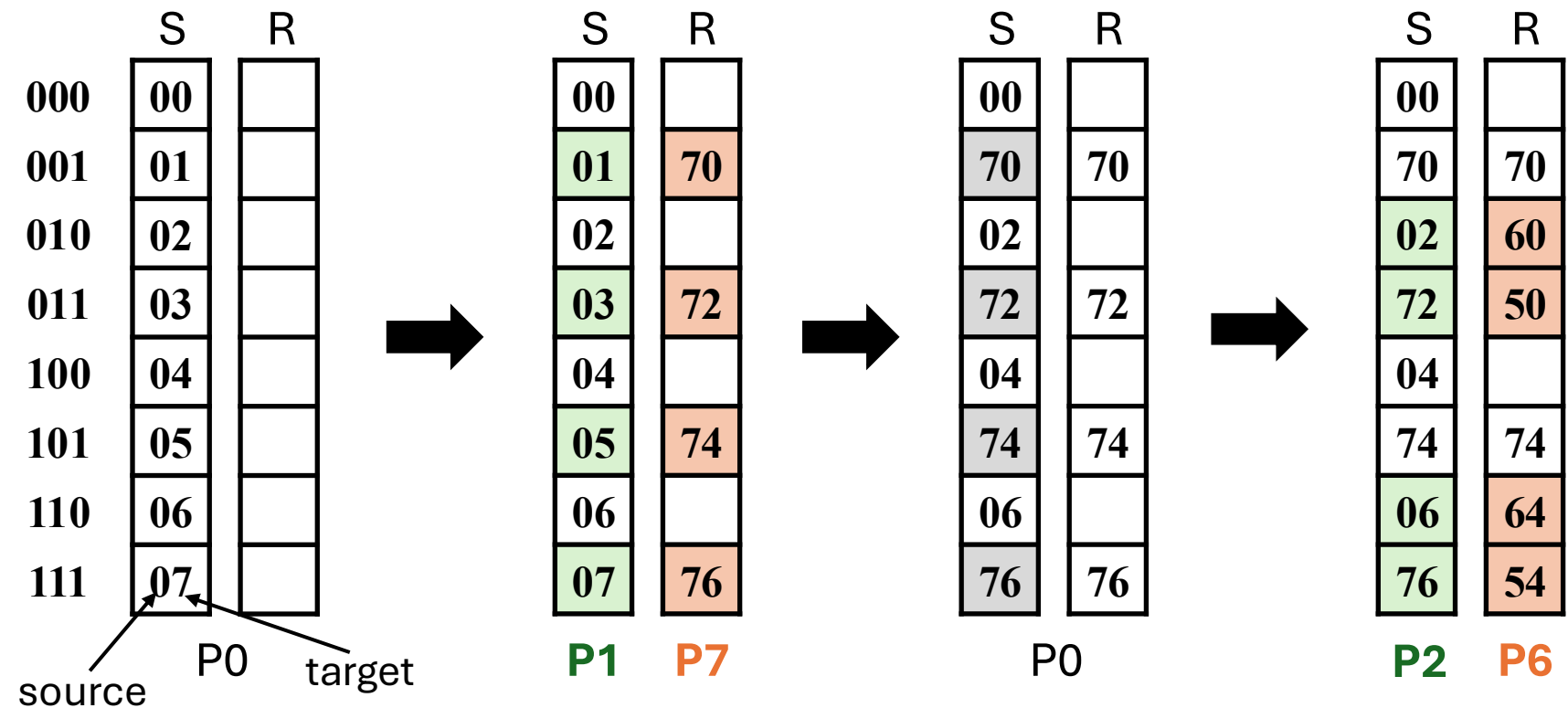
$$P = 8, r = 2$$

Bruck-style Algorithm Implies Blocking Communication



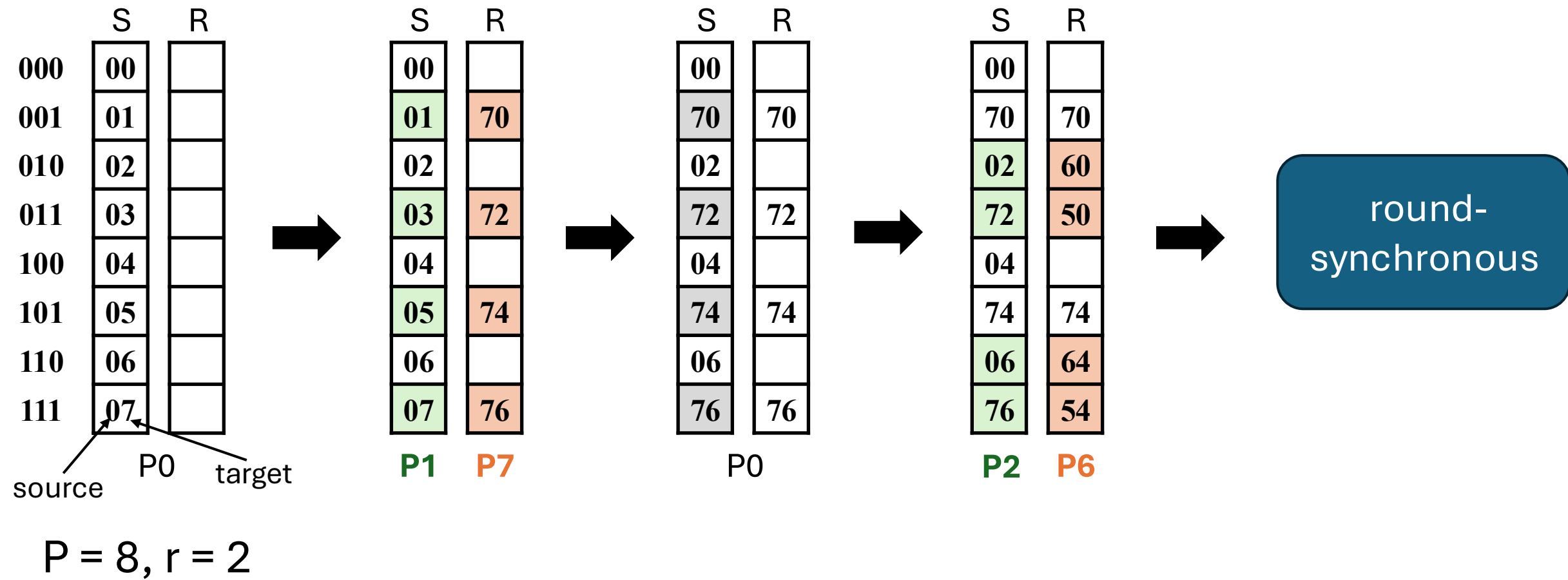
$$P = 8, r = 2$$

Bruck-style Algorithm Implies Blocking Communication



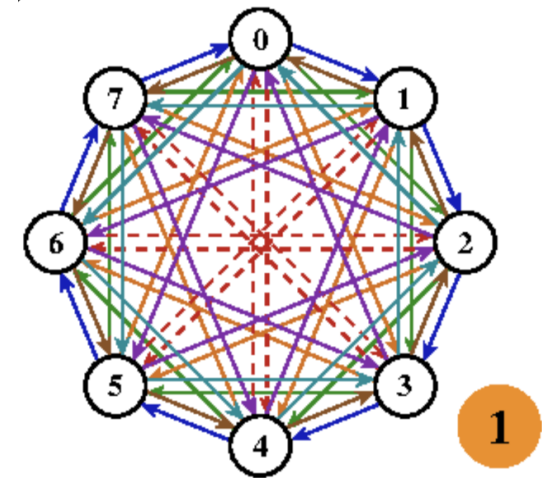
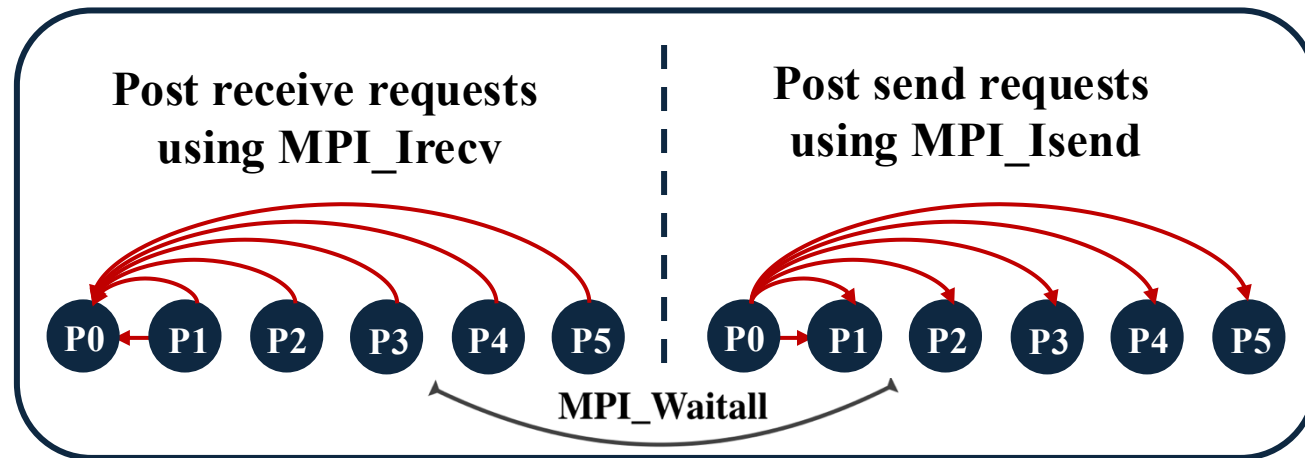
$$P = 8, r = 2$$

Bruck-style Algorithm Implies Blocking Communication



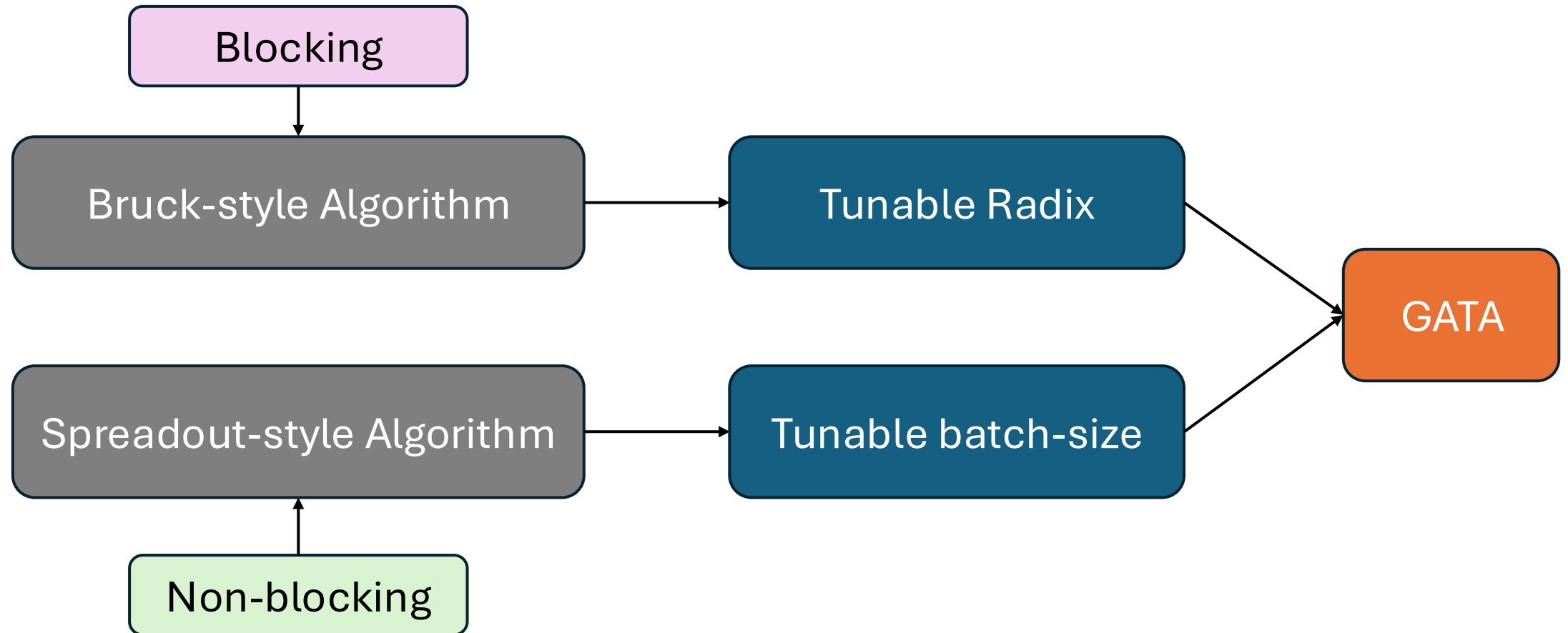
Spreadout-style Algorithm Implies Non-blocking Communication

Spread-out algorithm

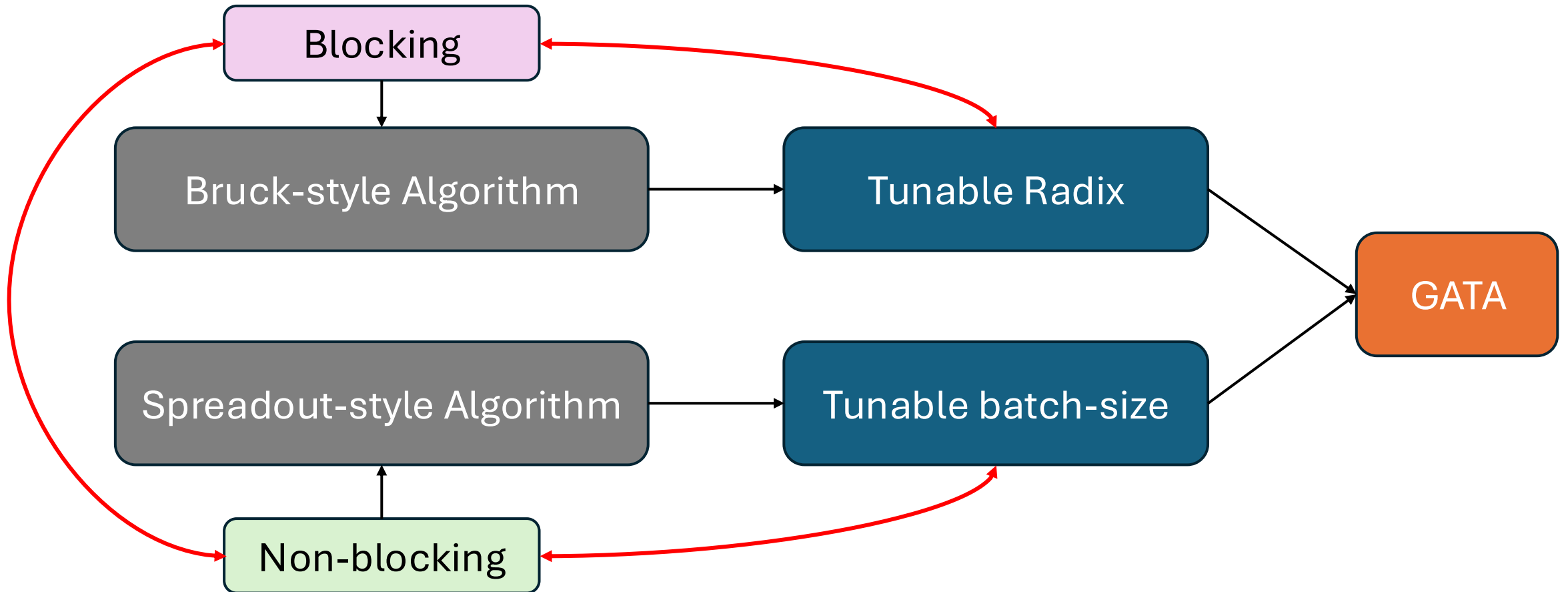


Non-blocking

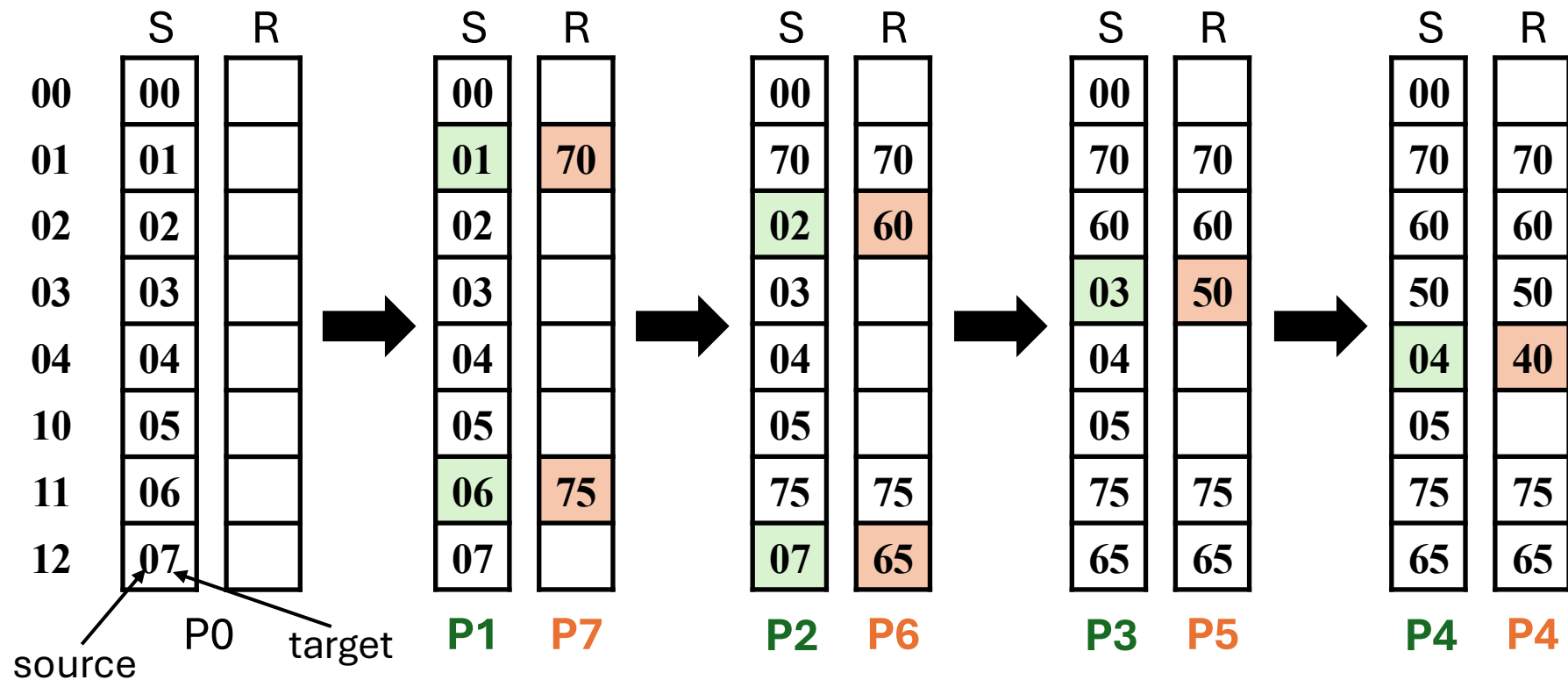
Generalized All-To-All Design



Generalized All-To-All Design

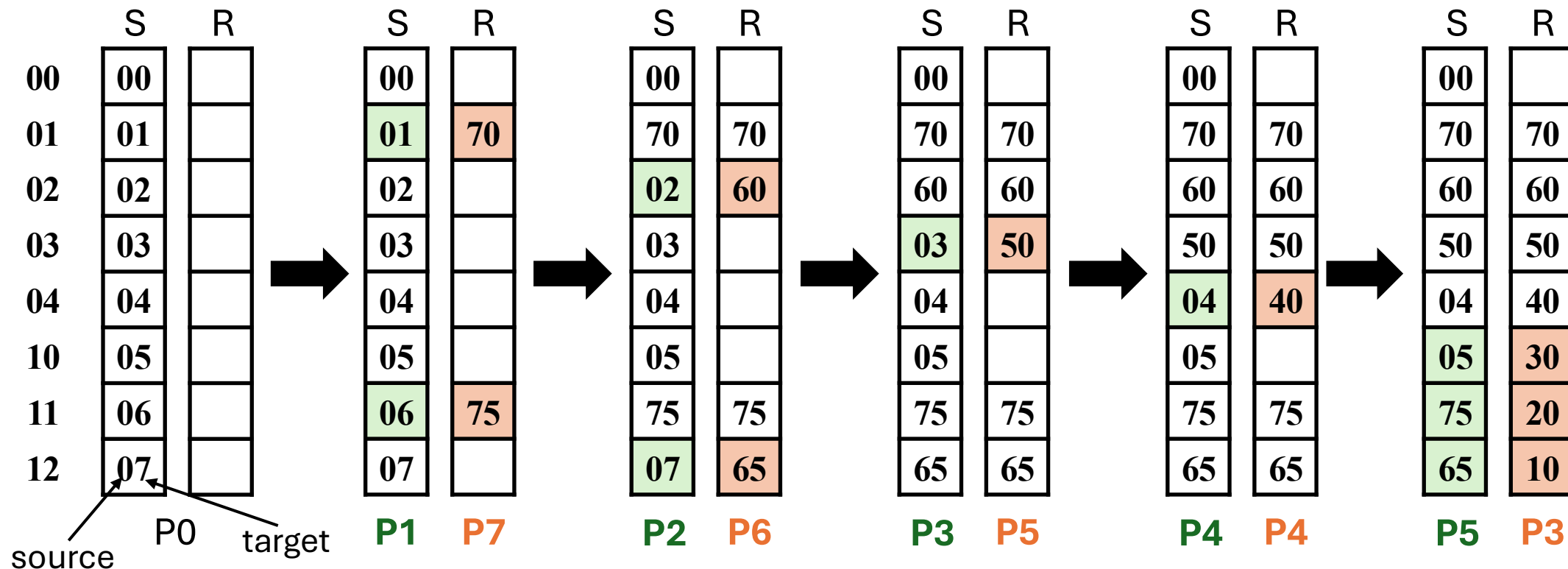


Tunable Radix All-to-all



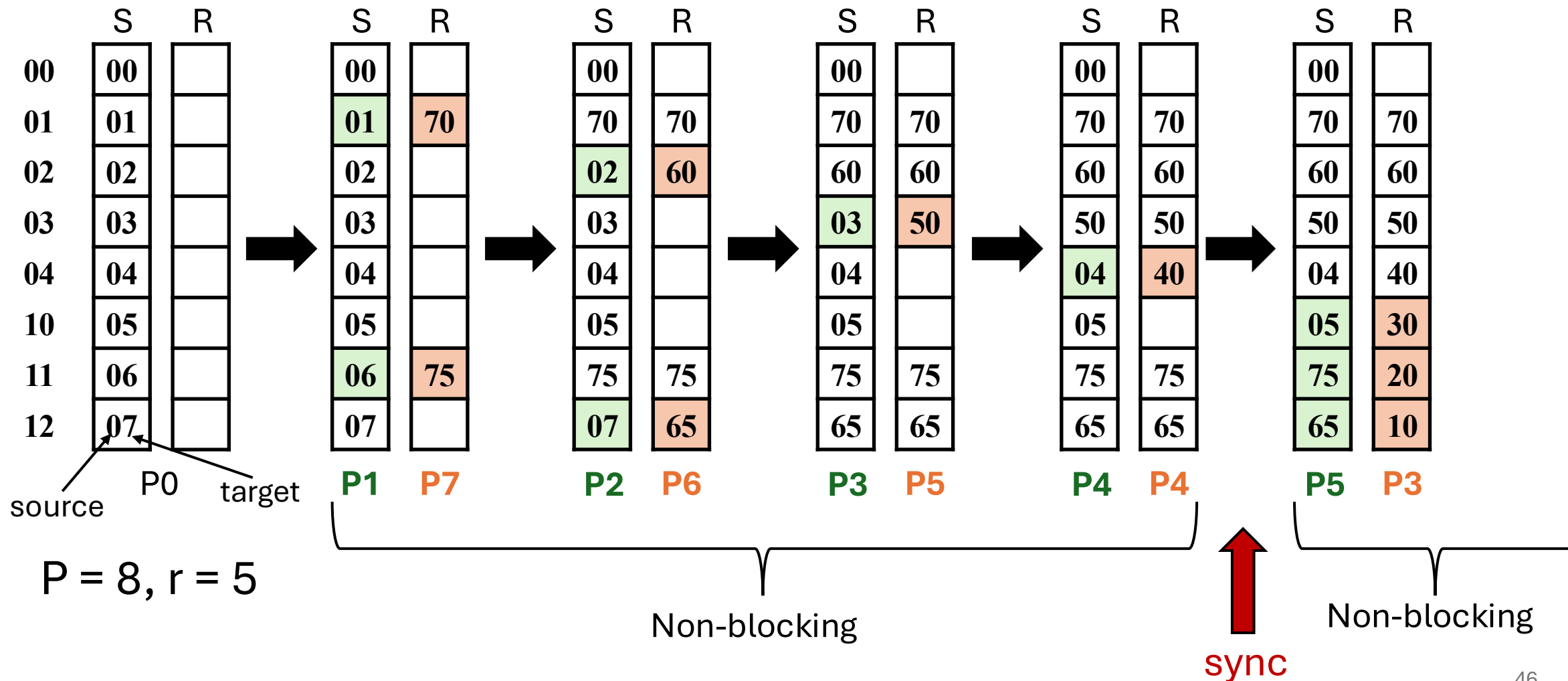
$$P = 8, r = 5$$

Tunable Radix All-to-all

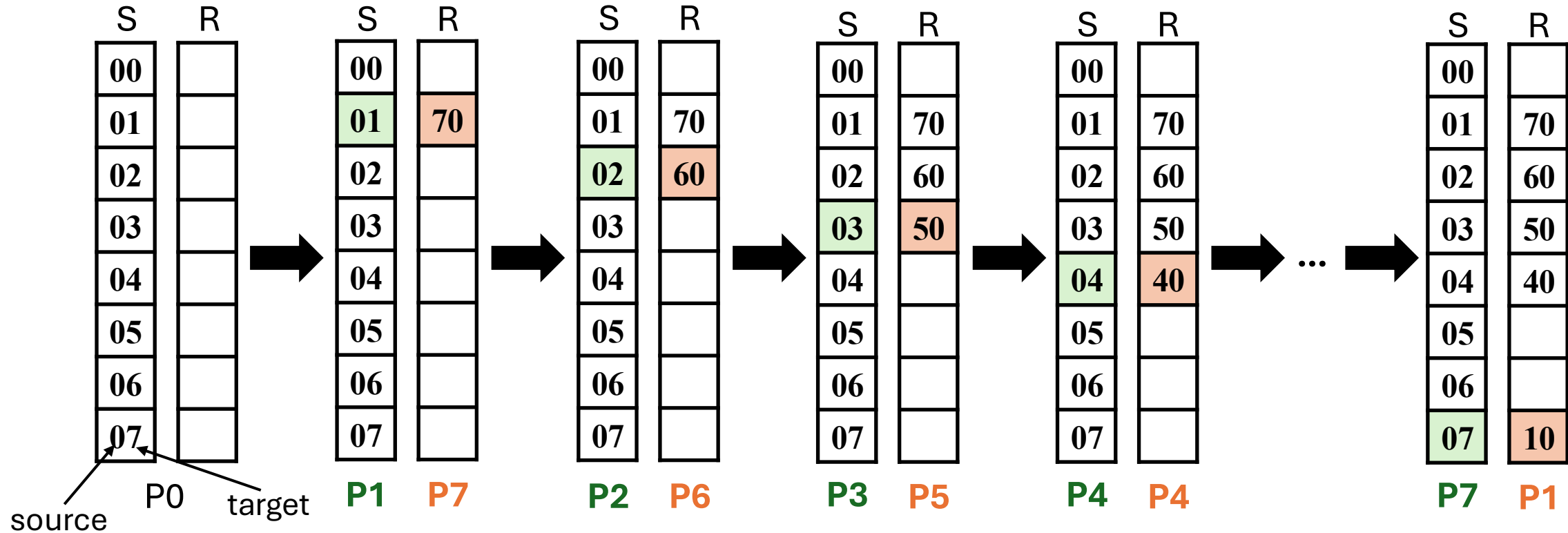


$$P = 8, r = 5$$

Tunable Radix All-to-all

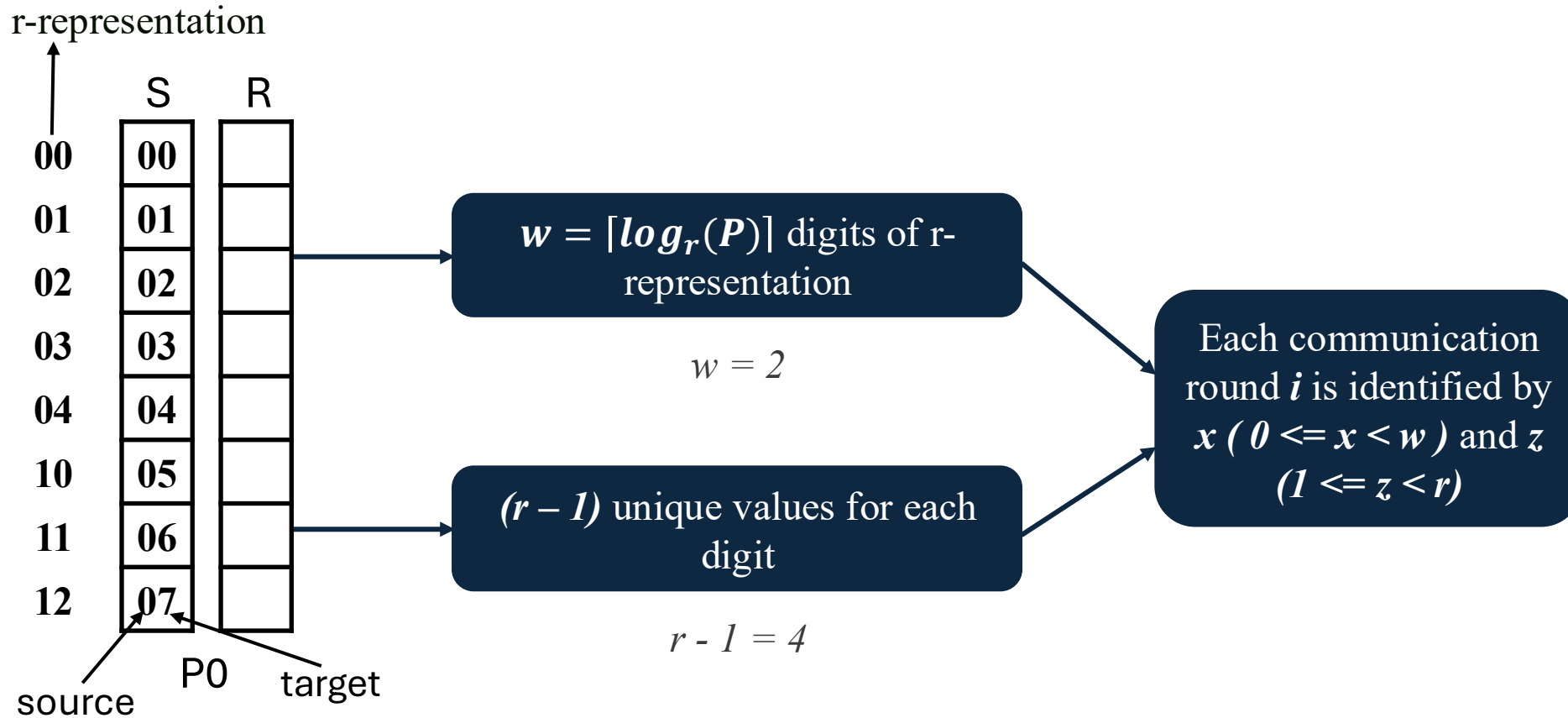


Extreme Radix ($r = P$)

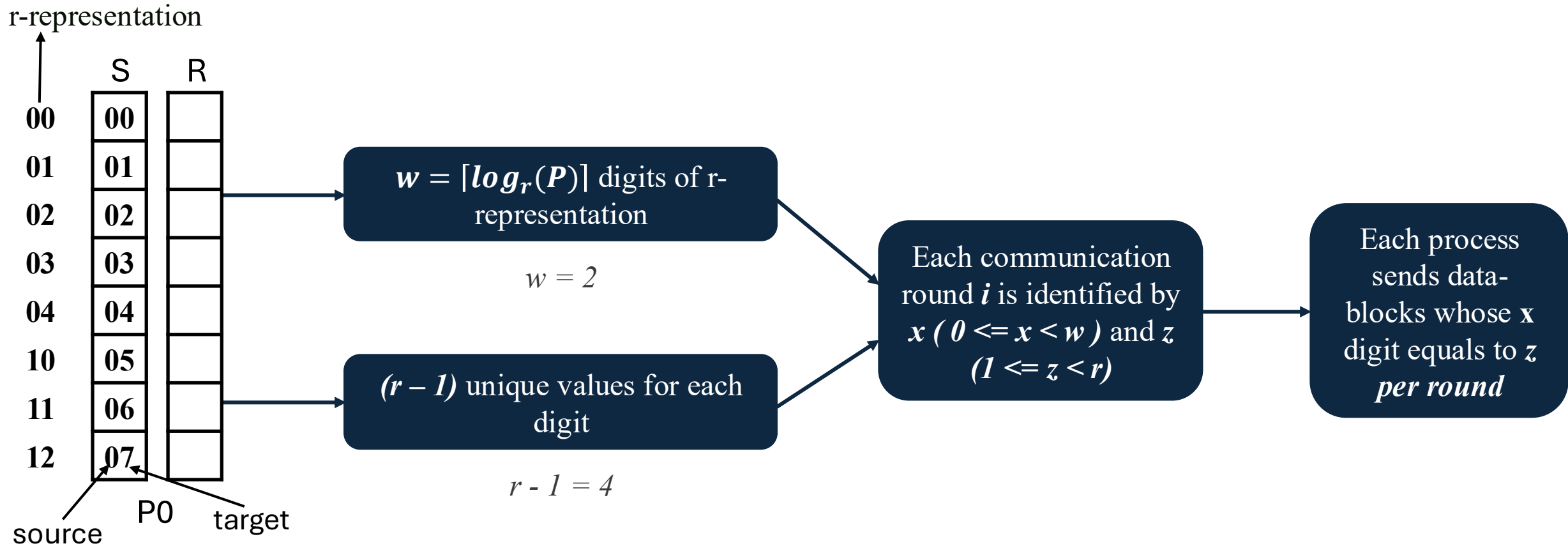


Non-blocking

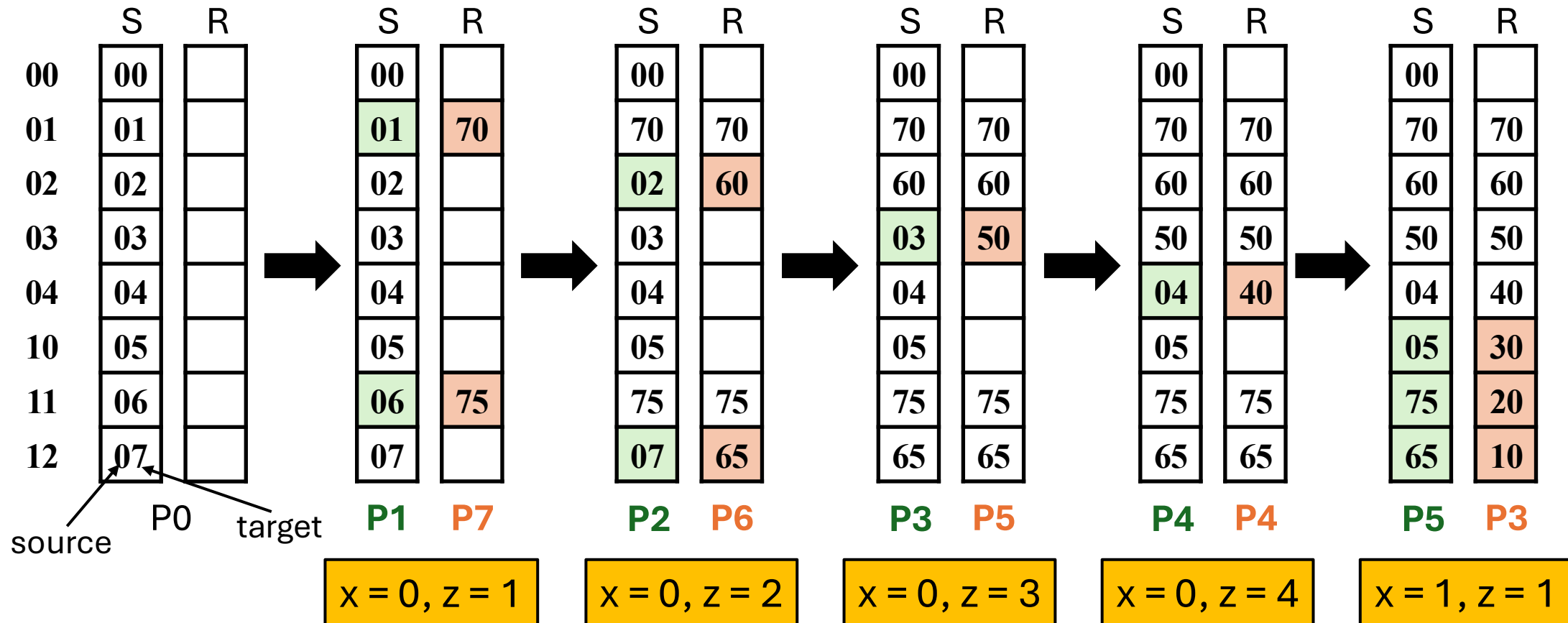
Tunable Radix All-to-all Definition



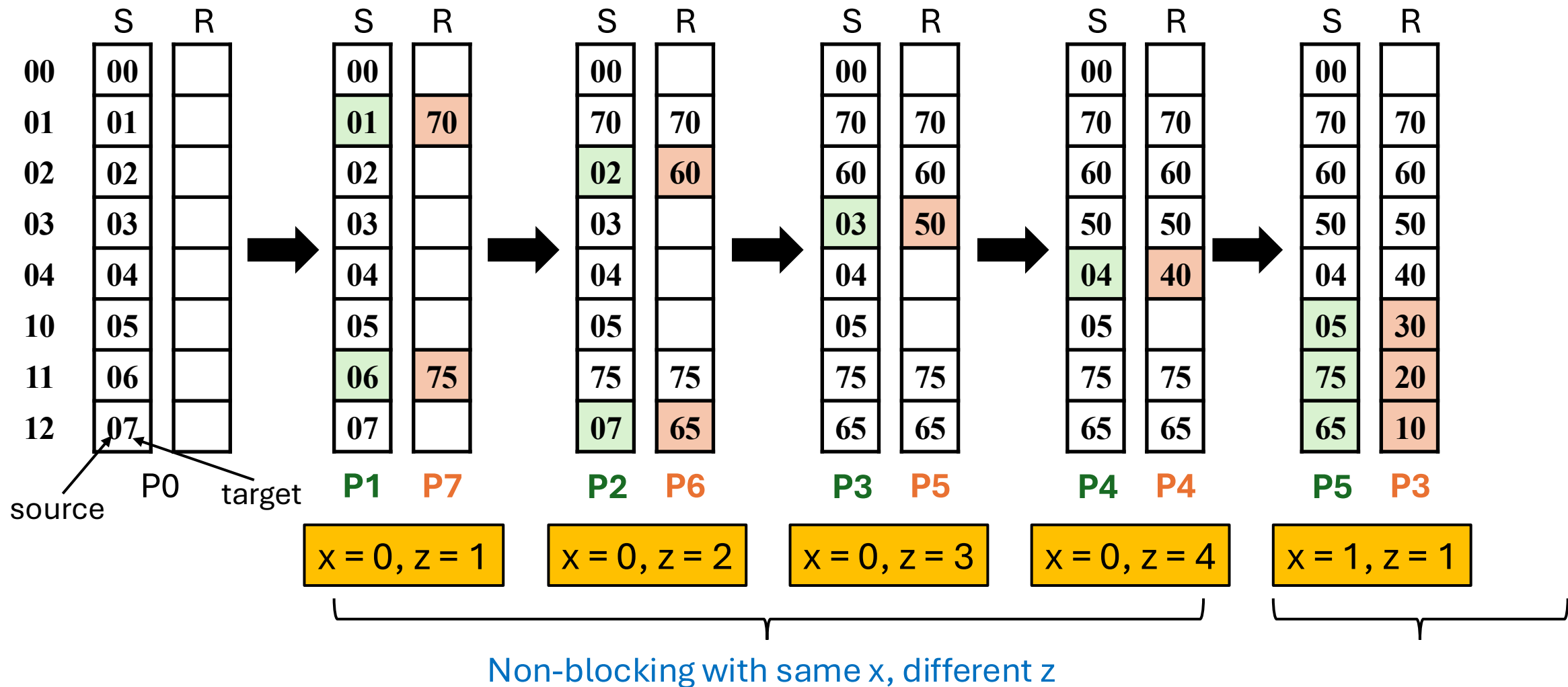
Tunable Radix All-to-all Definition



Tunable Radix All-to-all

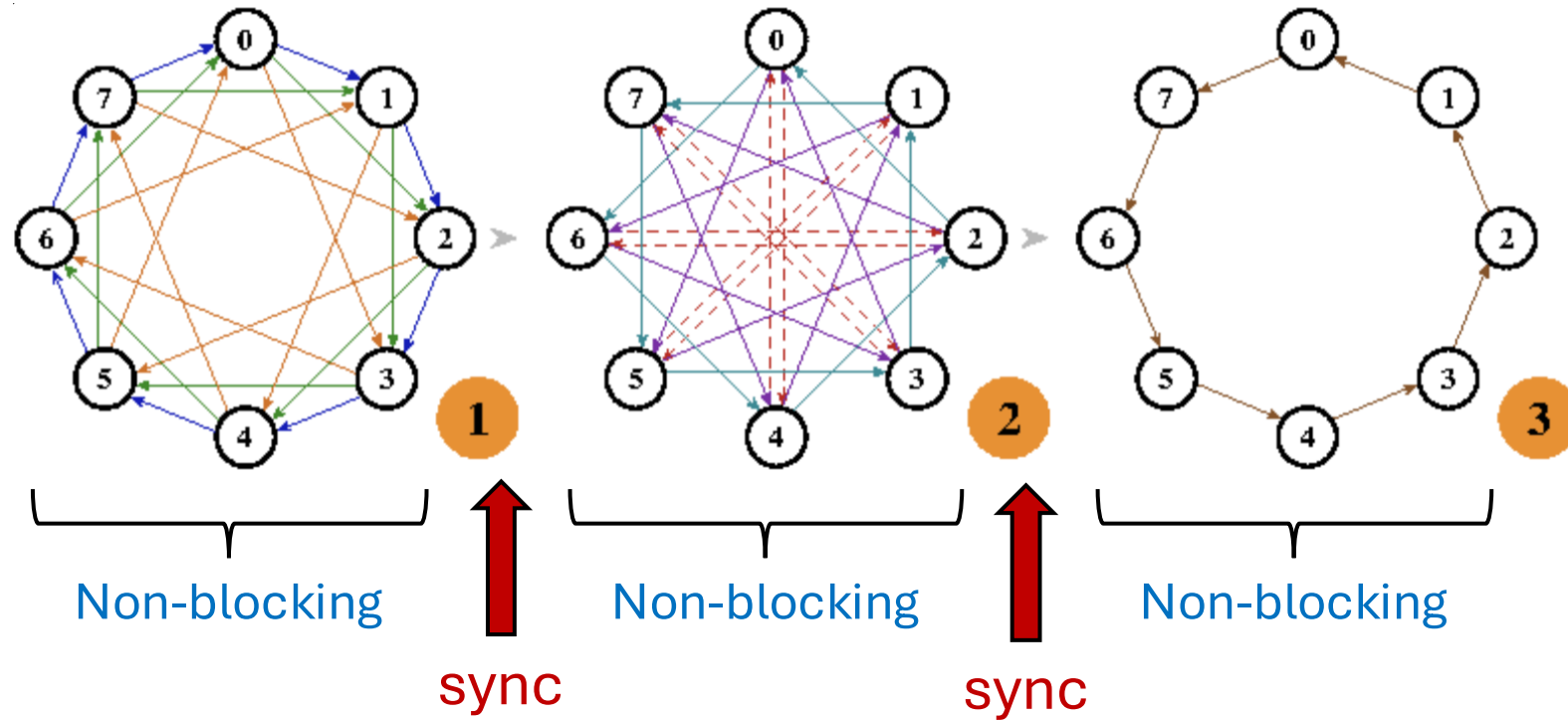


Tunable Radix All-to-all

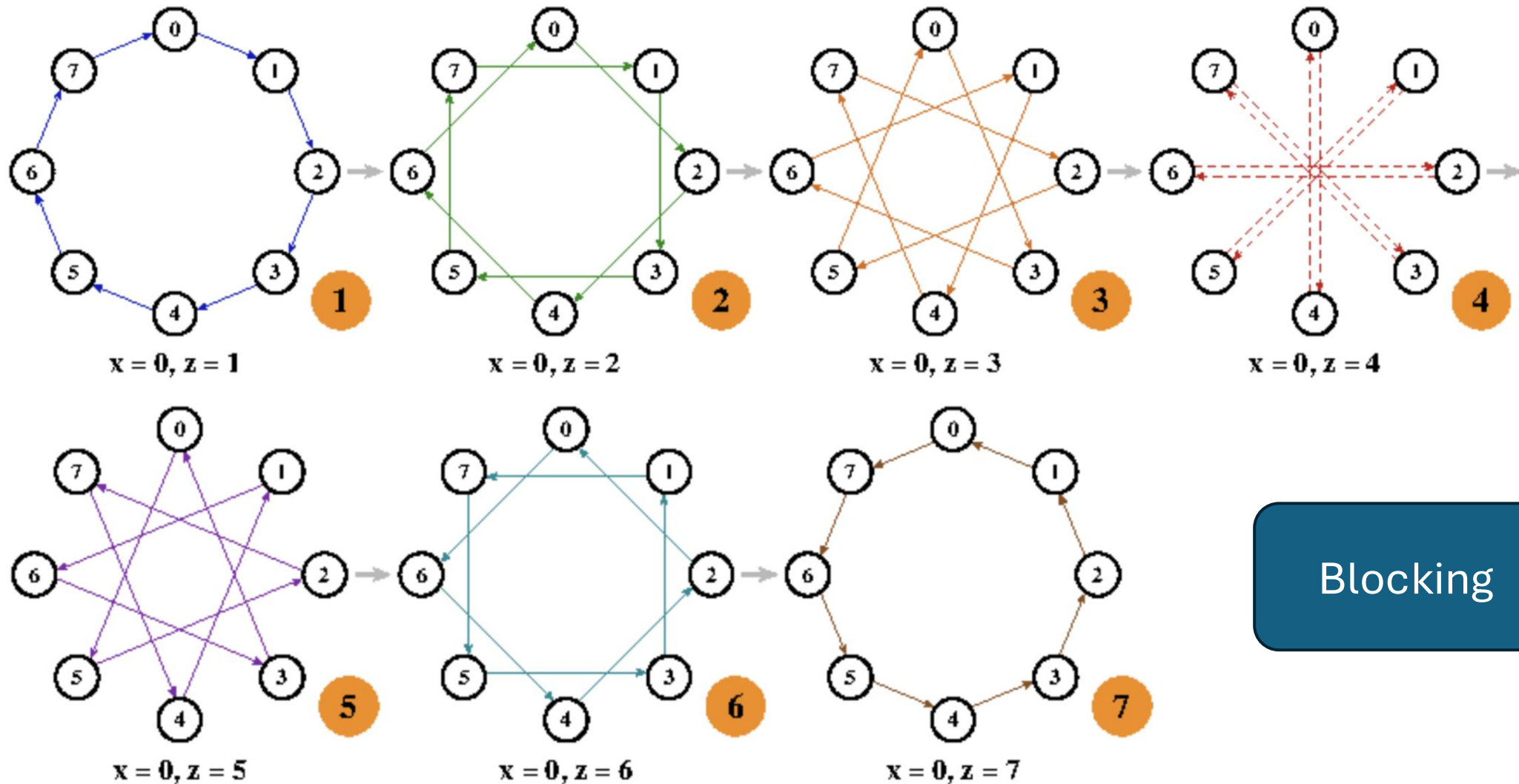


Tunable Batch-size b

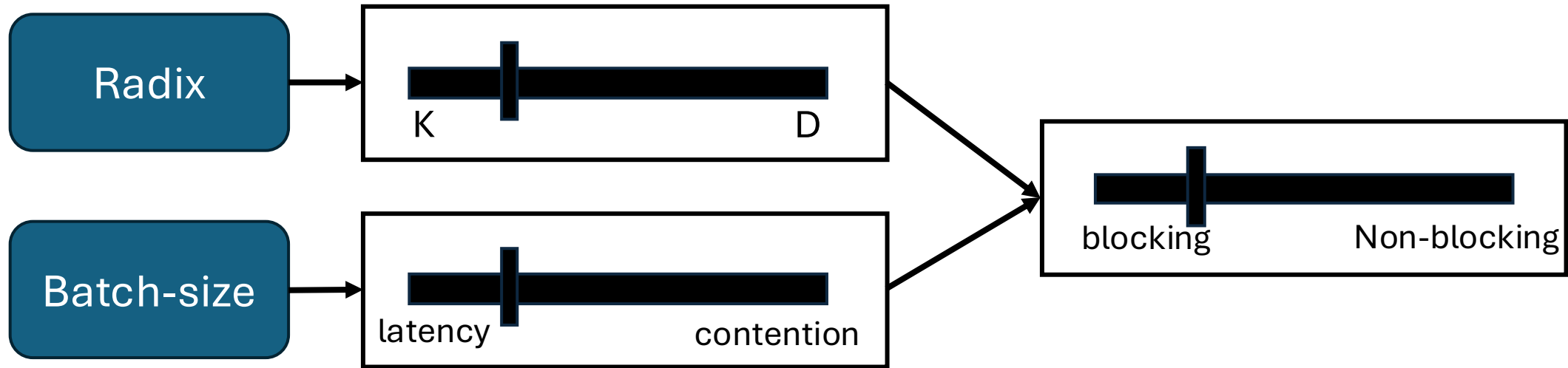
$b = 3$



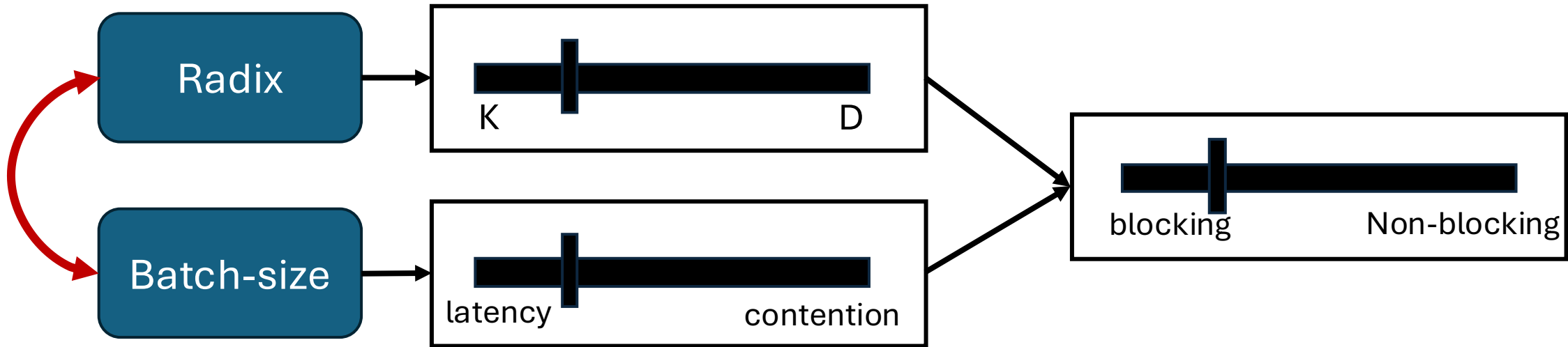
Extreme Batch-size ($b = 1$)



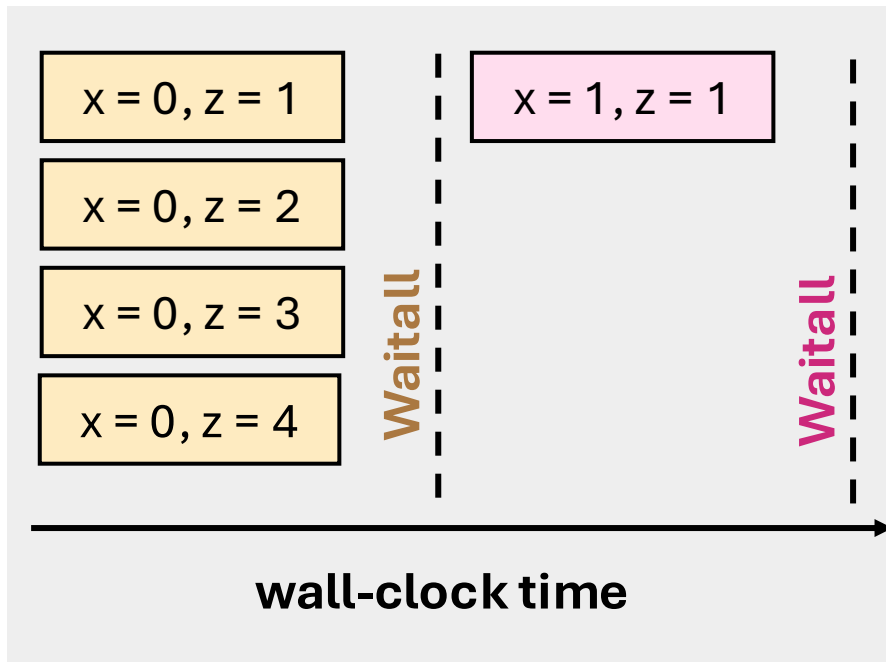
Joint (r,b) Enables Dense 2D Tunable Space



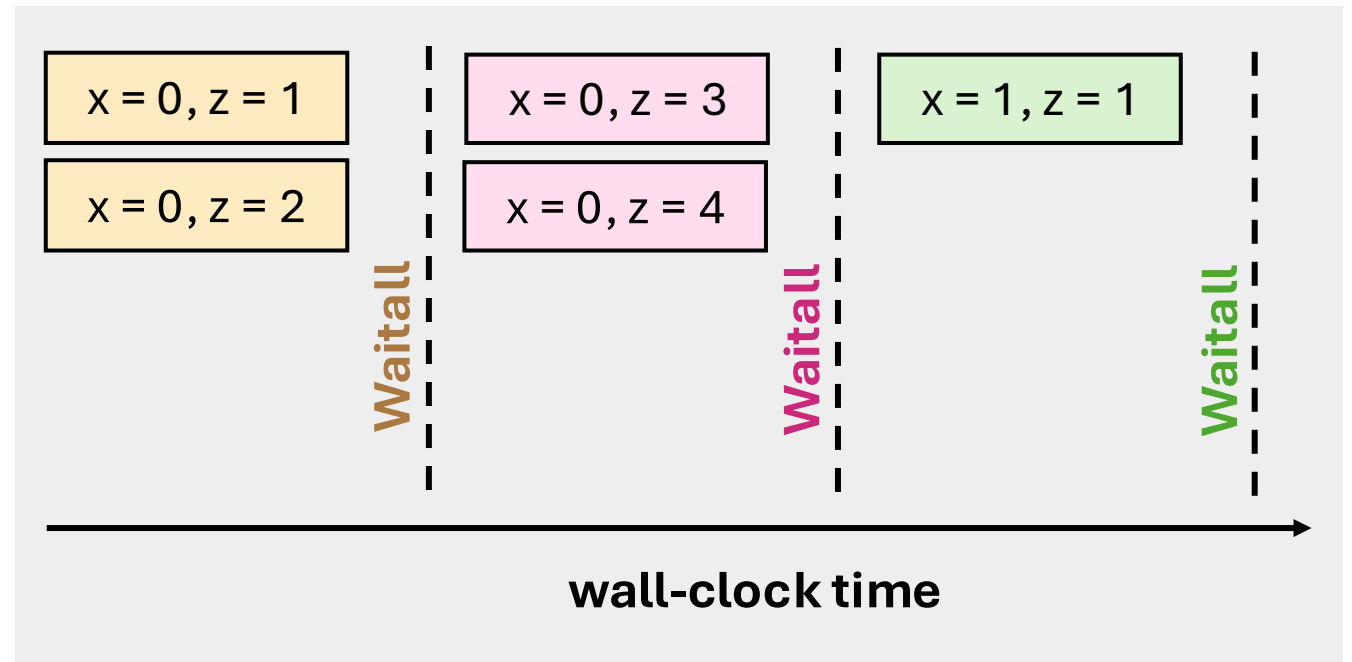
Joint (r,b) Enables Dense 2D Tunable Space



(r,b) Design Space With 8 Processes

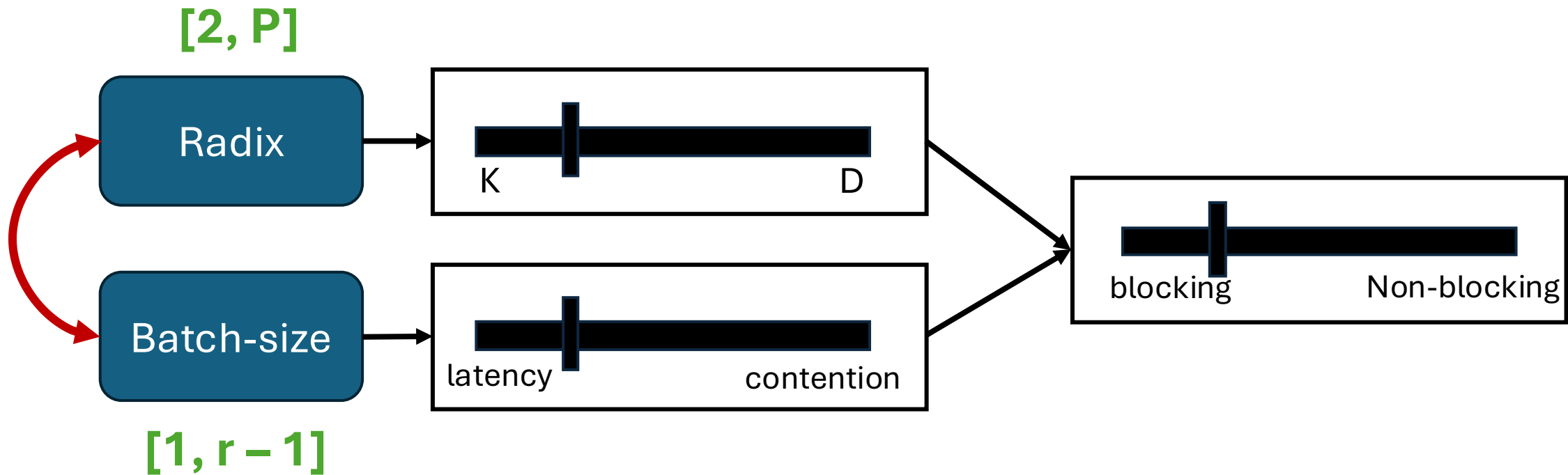


$$P = 8, r = 5, b = 4$$

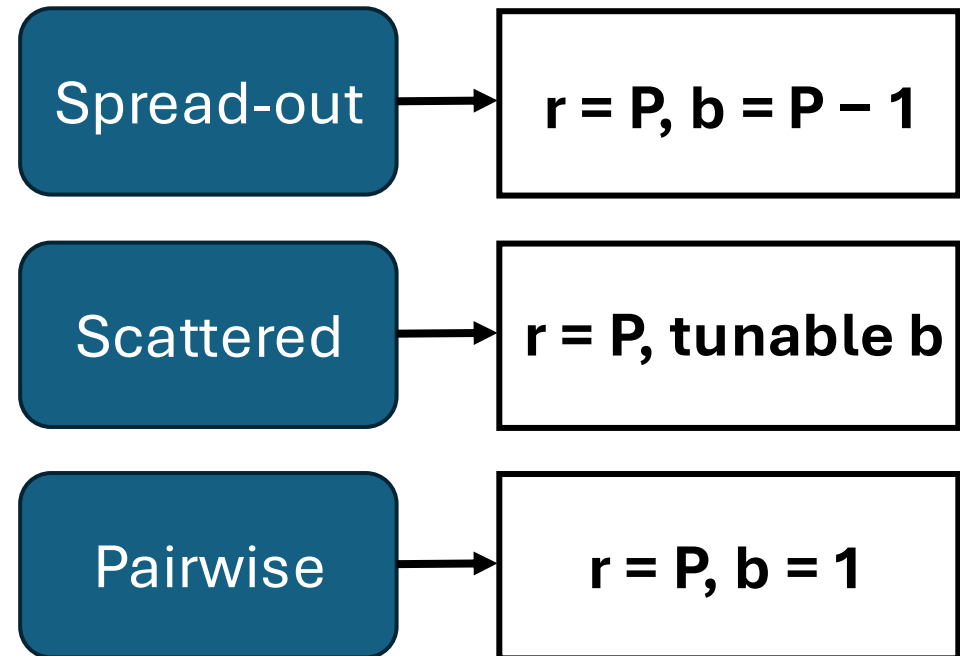
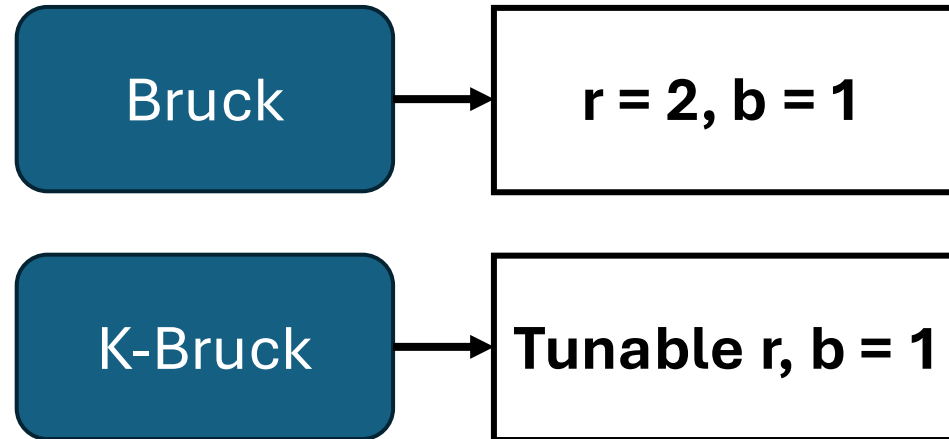


$$P = 8, r = 5, b = 2$$

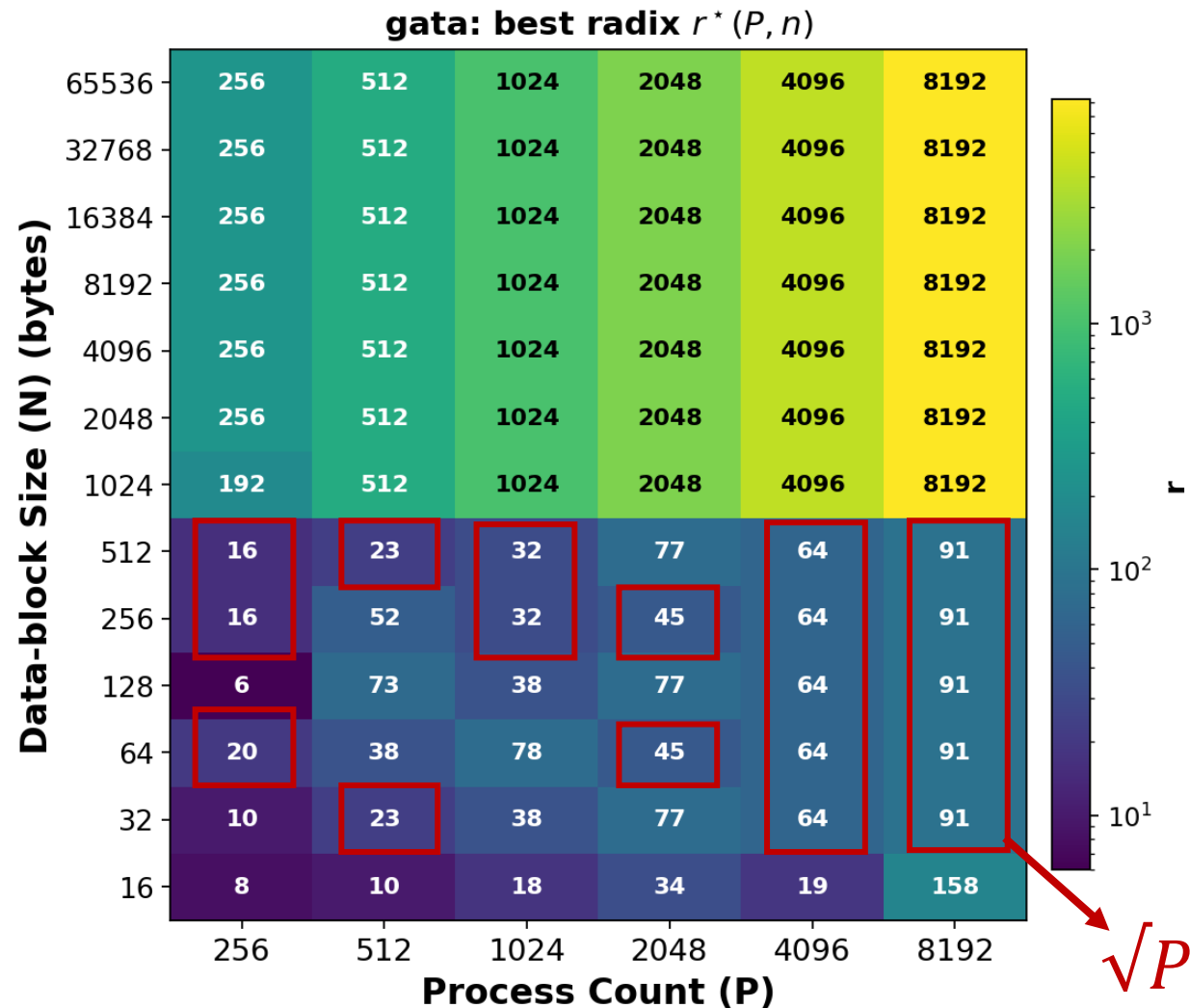
Joint (r,b) Enables Dense 2D Tunable Space



Boundary Cases

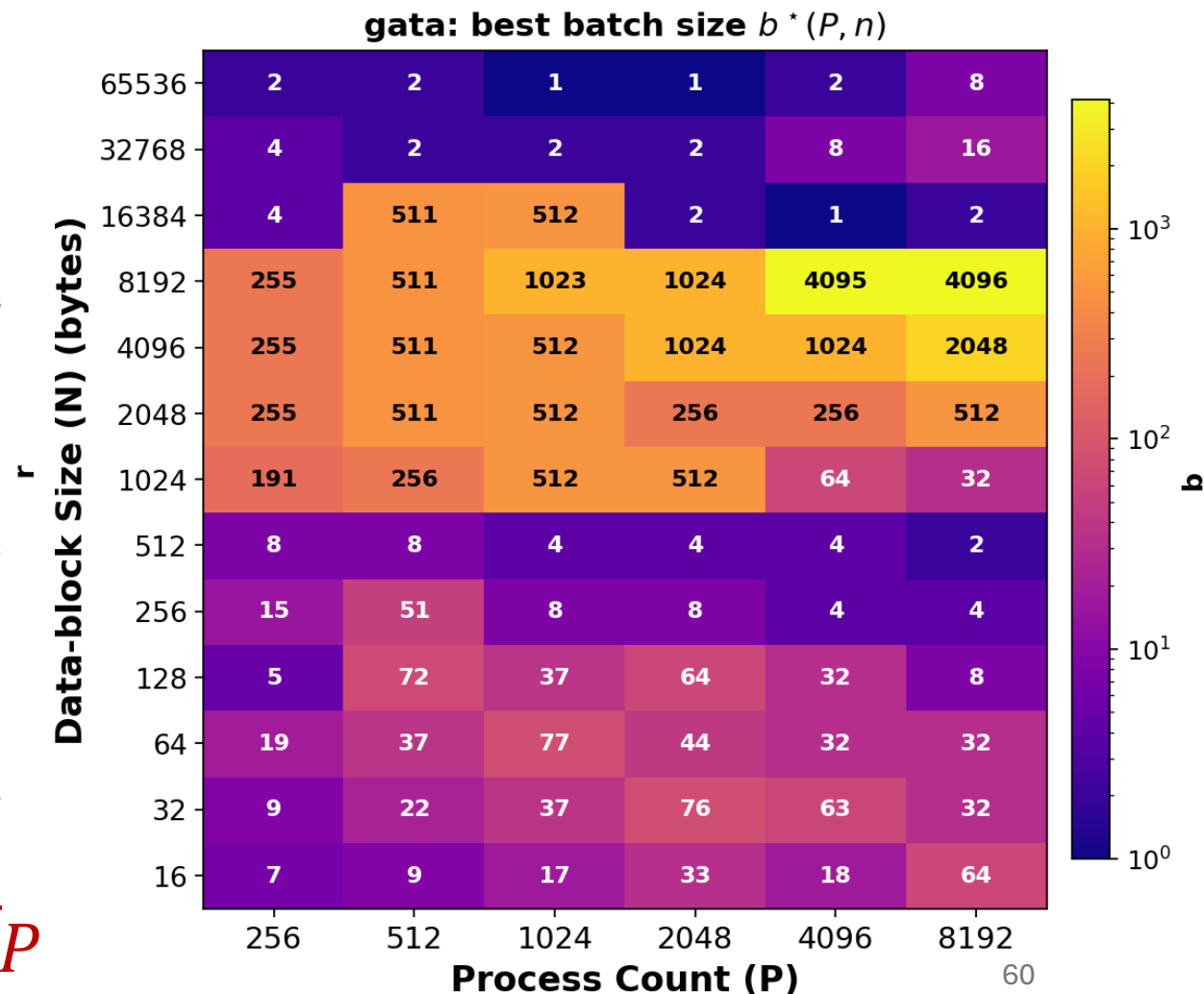
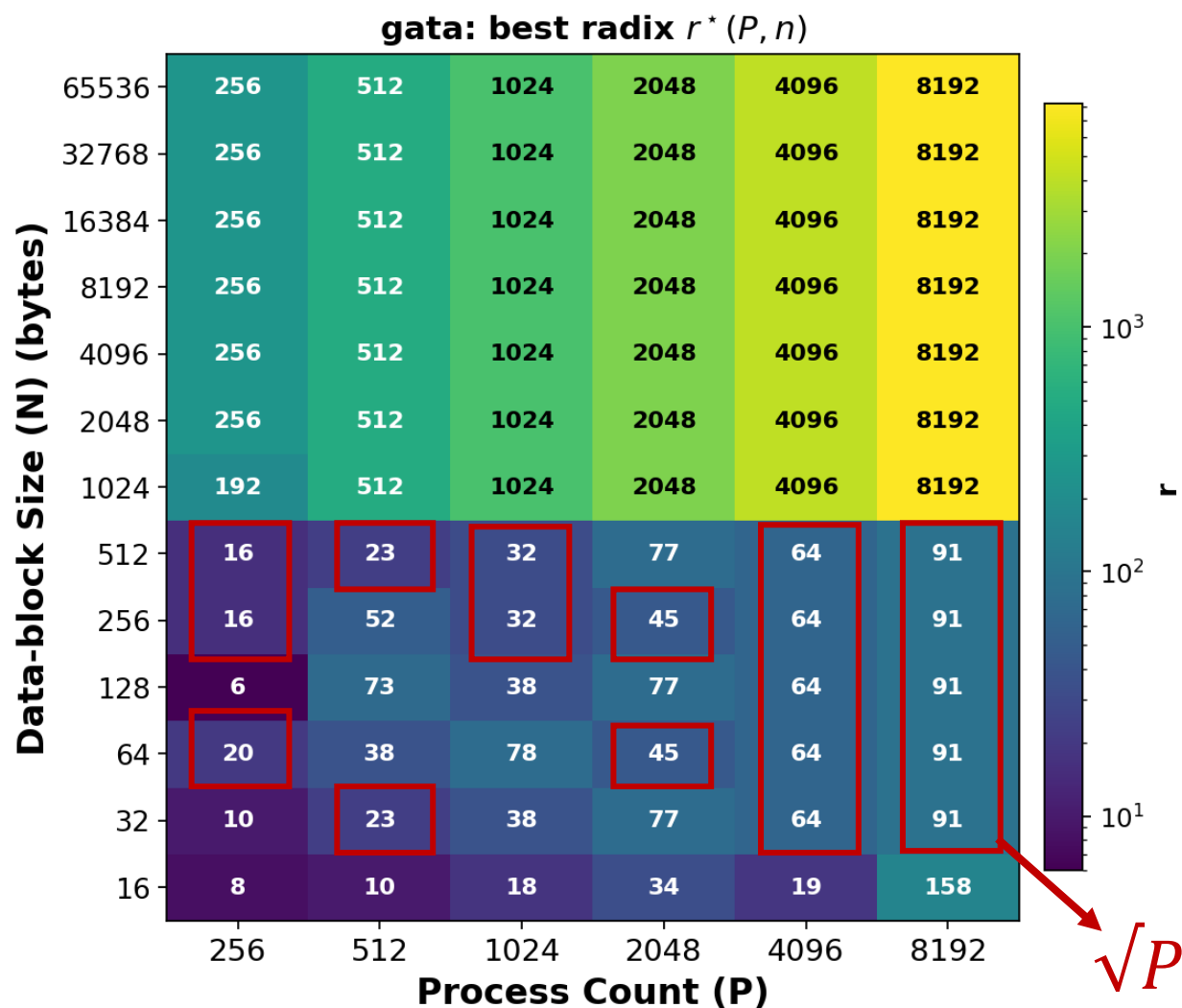


Evaluation: Tunable Radix Selection

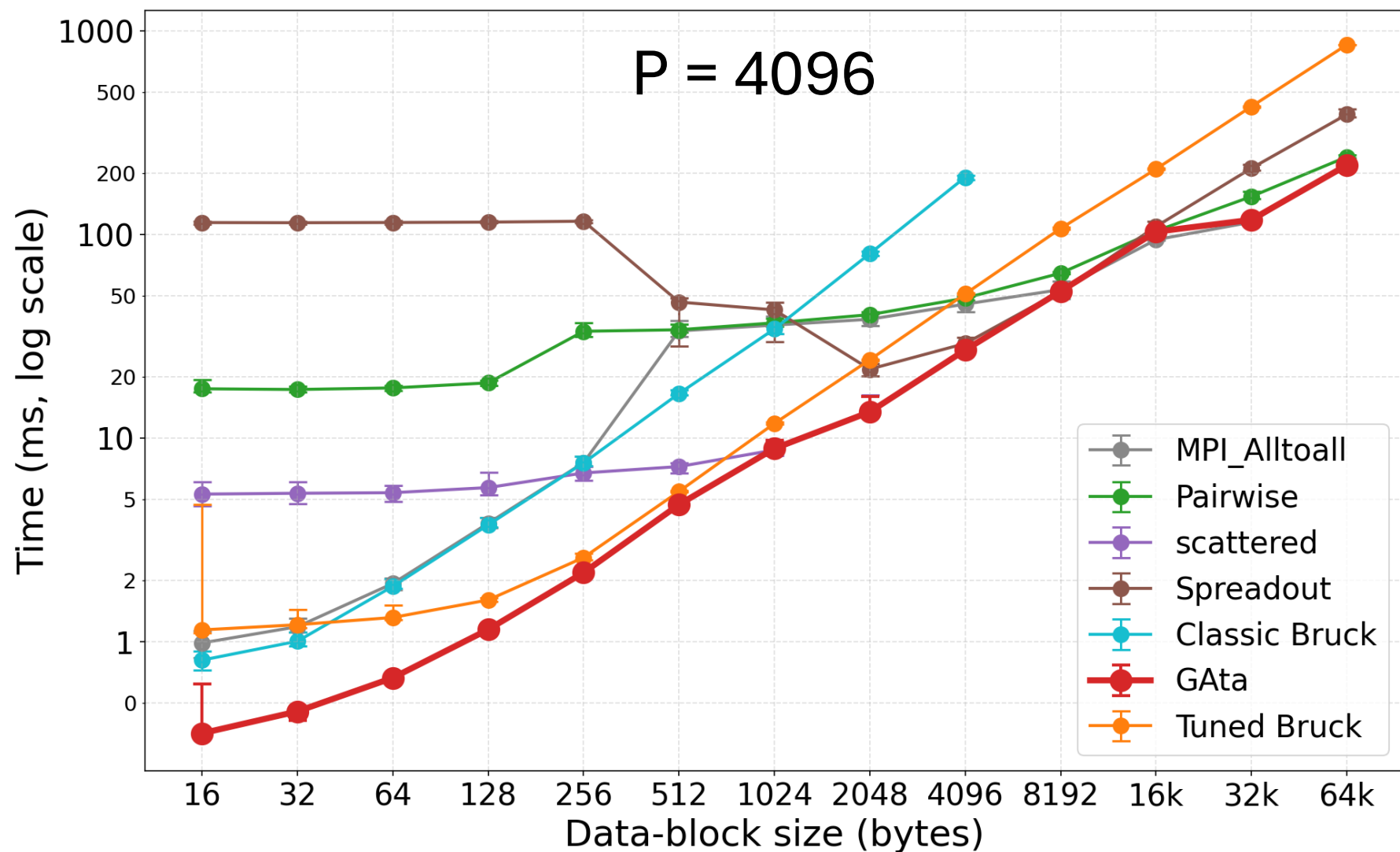


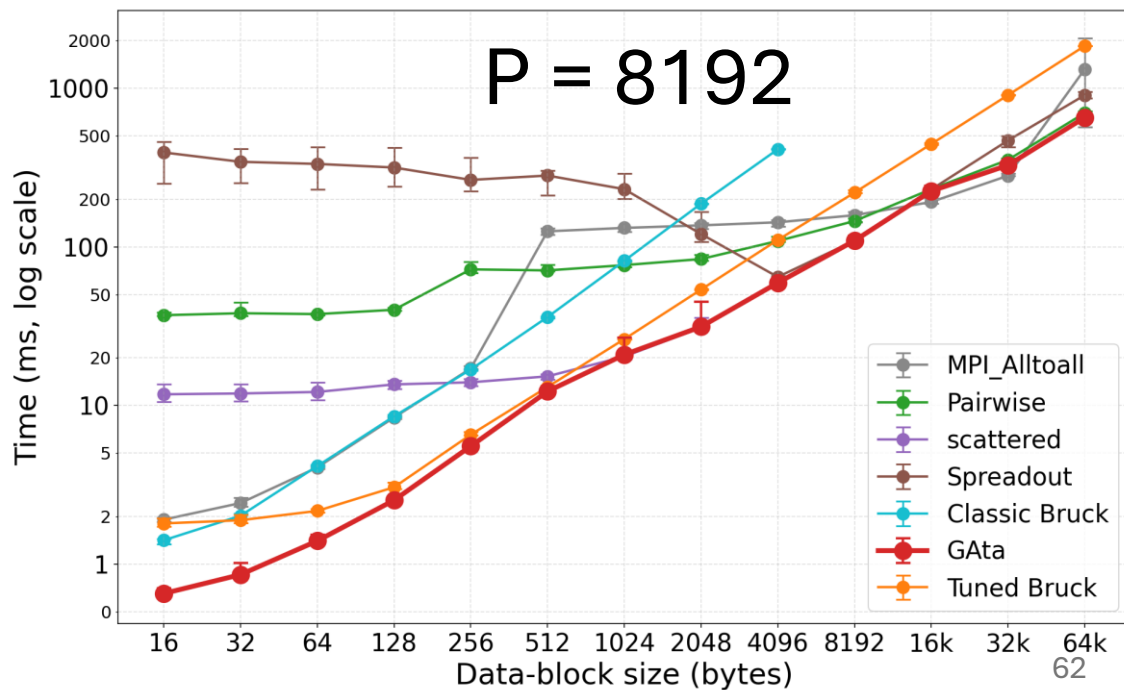
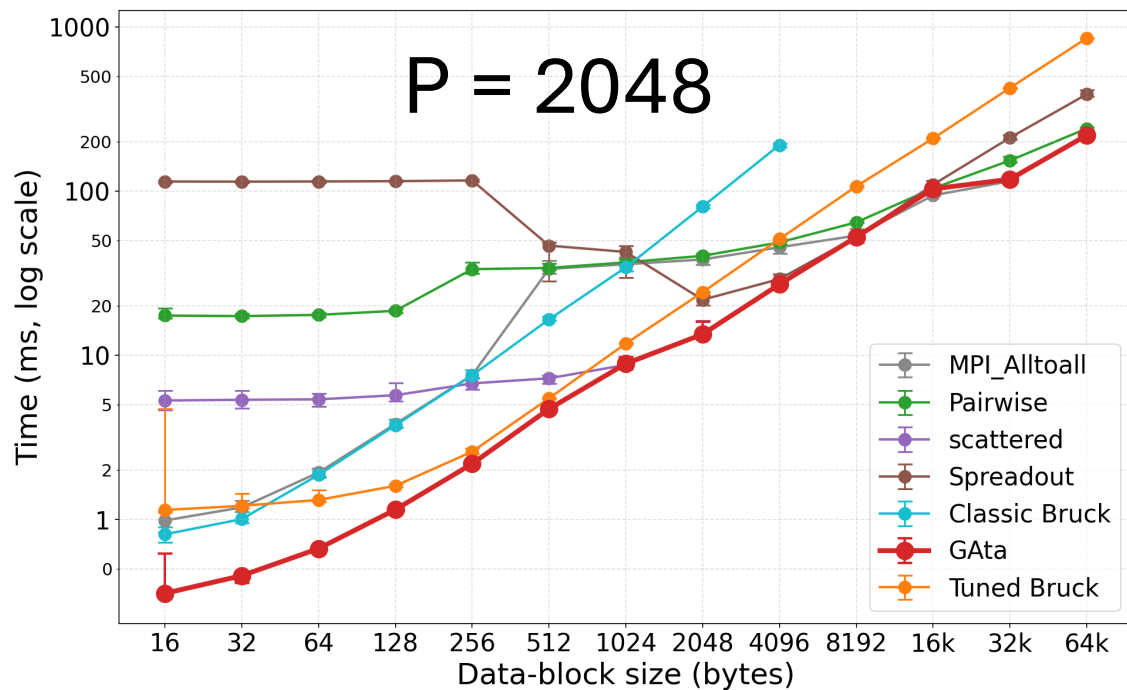
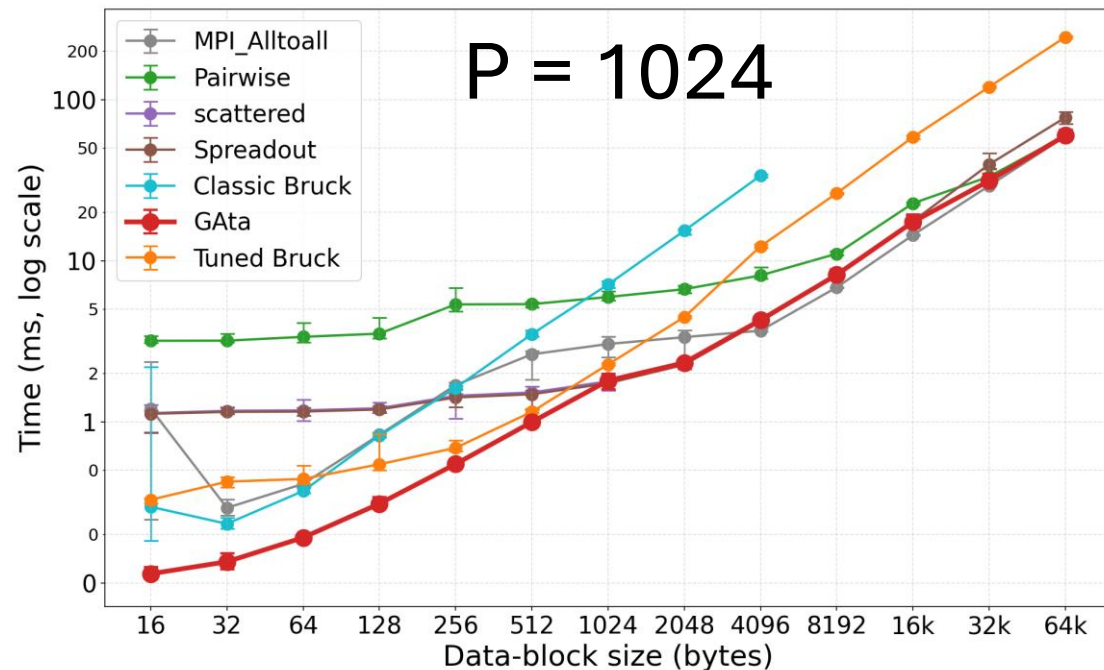
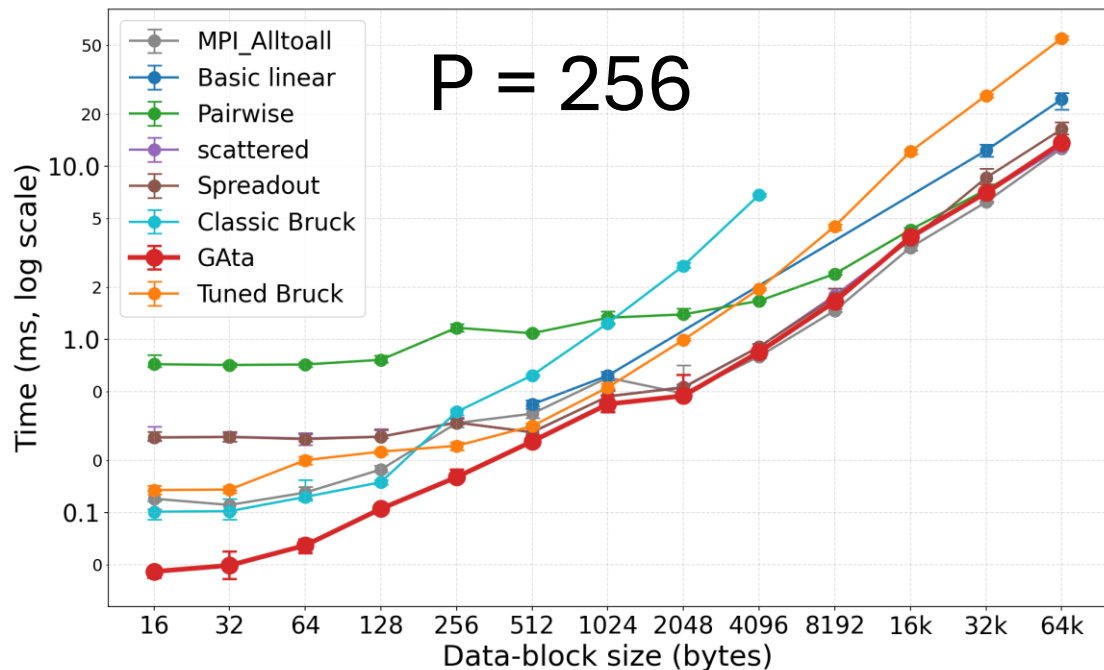
Polaris Supercomputer at
Argonne National Laboratory

Evaluation: Tunable Batch-size Selection

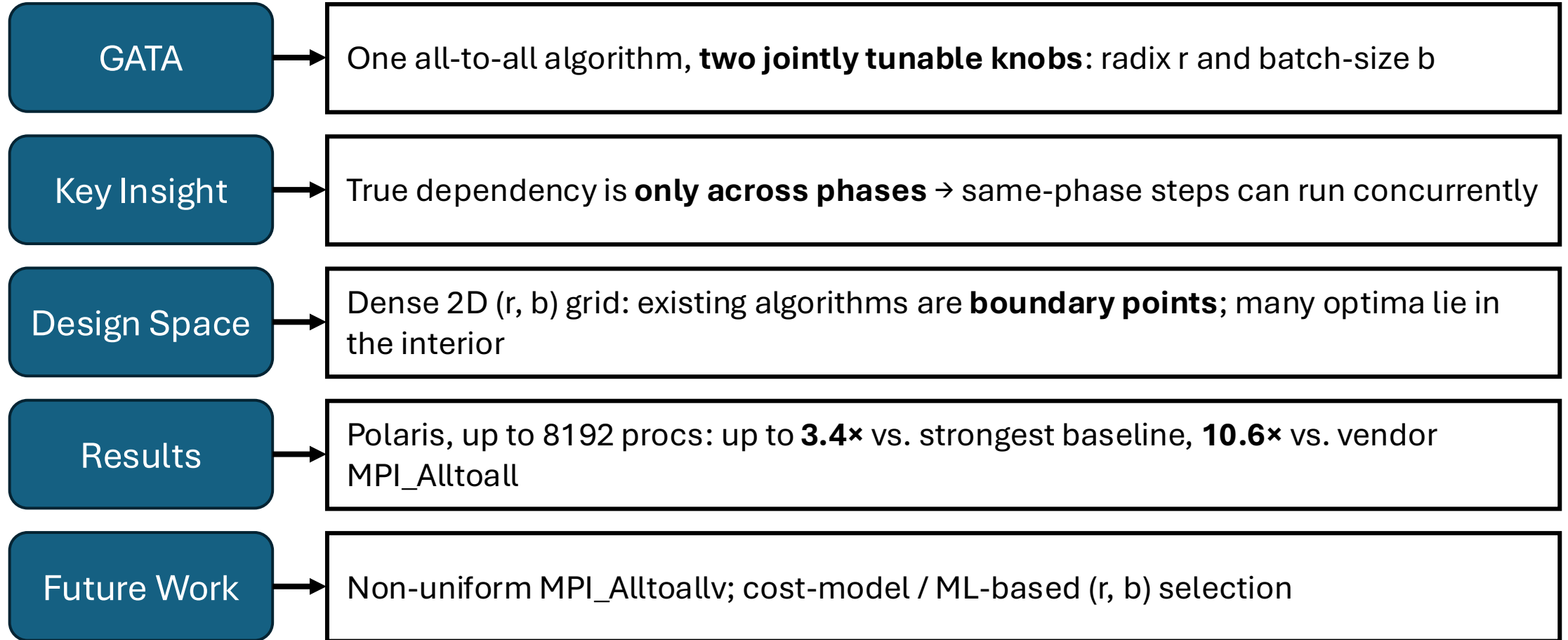


Performance Comparison





Conclusion



Thank you!