Parameterized Algorithms for Non-uniform All-to-all

Presenter: Jens Domke

Authors: Ke Fan, Jens Domke, Seydou Ba, and Sidharth Kumar

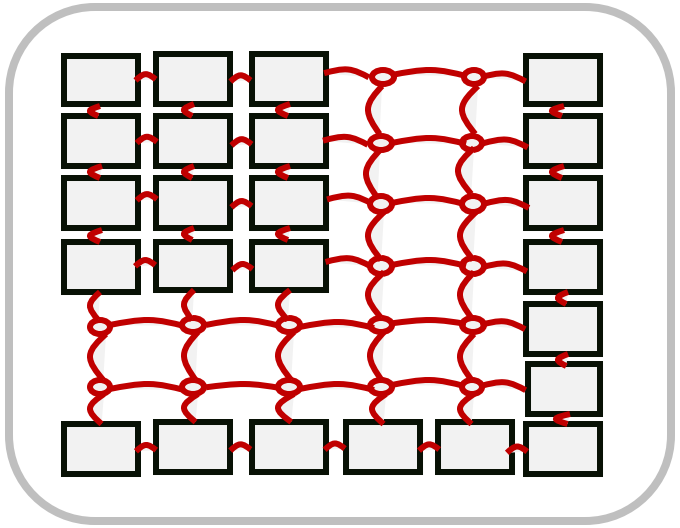
Agenda

- Introduction and Background
  - Inter-process Communication
  - All-to-all Data Exchange
  - Standard MPI All-to-all Implementations
- Challenges and Solutions
  - Tunability of Performance
  - Logarithmic Non-uniform All-to-all
- Parameterized Logarithmic Algorithm
- Parameterized Linear Algorithm
- Evaluation
- Application
- Conclusion

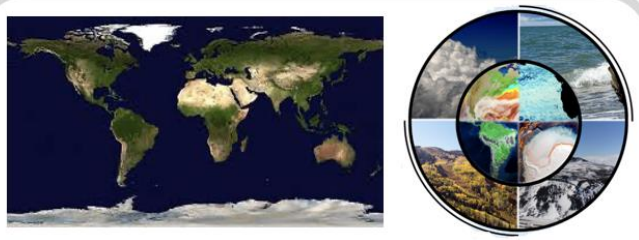
Agenda

- Introduction and Background
  - Inter-process Communication
  - All-to-all Data Exchange
  - Standard MPI All-to-all Implementations
- Challenges and Solutions
  - Tunability of Performance
  - Logarithmic Non-uniform All-to-all
- Parameterized Logarithmic Algorithm
- Parameterized Linear Algorithm
- Evaluation
- Application
- Conclusion

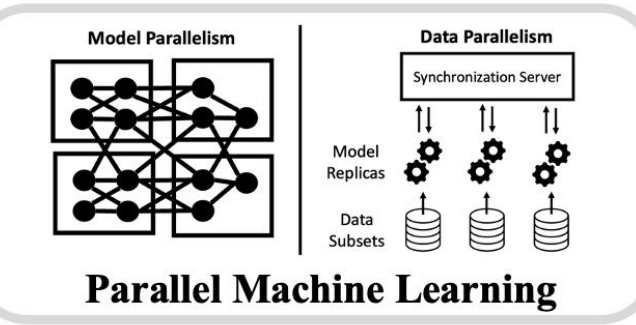
Exascale Computing Necessaries Effective Parallelism



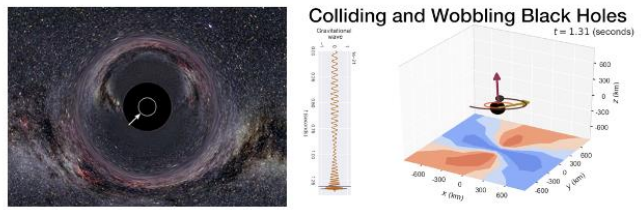
Exascale Supercomputers



Earth and Climate Modeling



Parallel Machine Learning

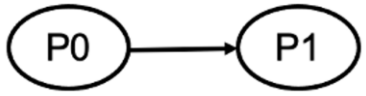


Black Holes Simulation

Inter-process Data Movement

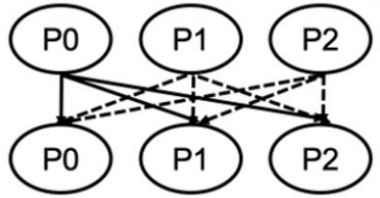
Three Fundamental Interfaces of Data Movement

Point-to-point

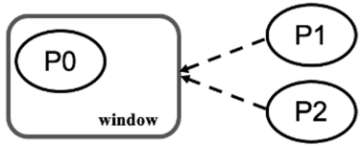


MPI_Send
MPI_Recv

Collective



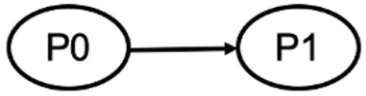
One-sided



MPI_Put
MPI_Get

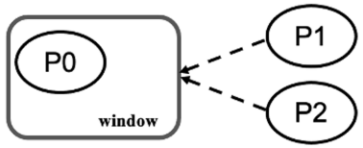
Collective Involves Communication Among All Processes

Point-to-point



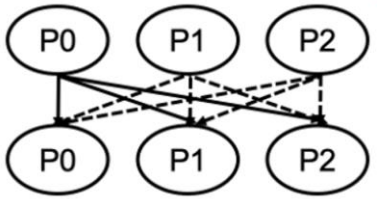
MPI_Send
MPI_Recv

One-sided

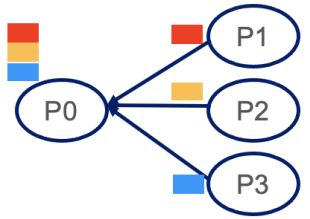


MPI_Put
MPI_Get

Collective

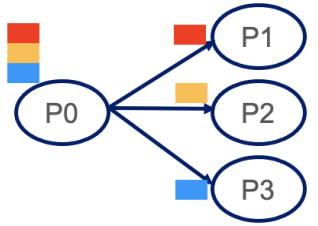


All-to-One



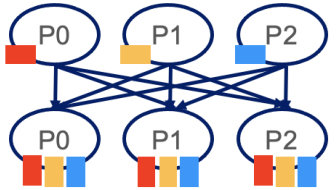
e.g., MPI_Reduce

One-to-All



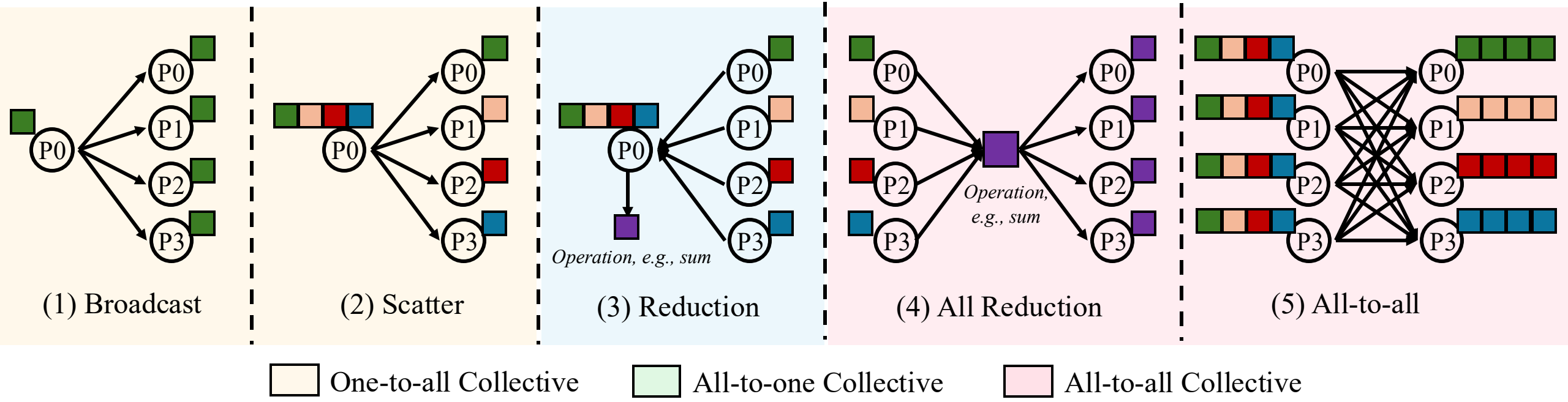
e.g., MPI_Scatter

All-to-All

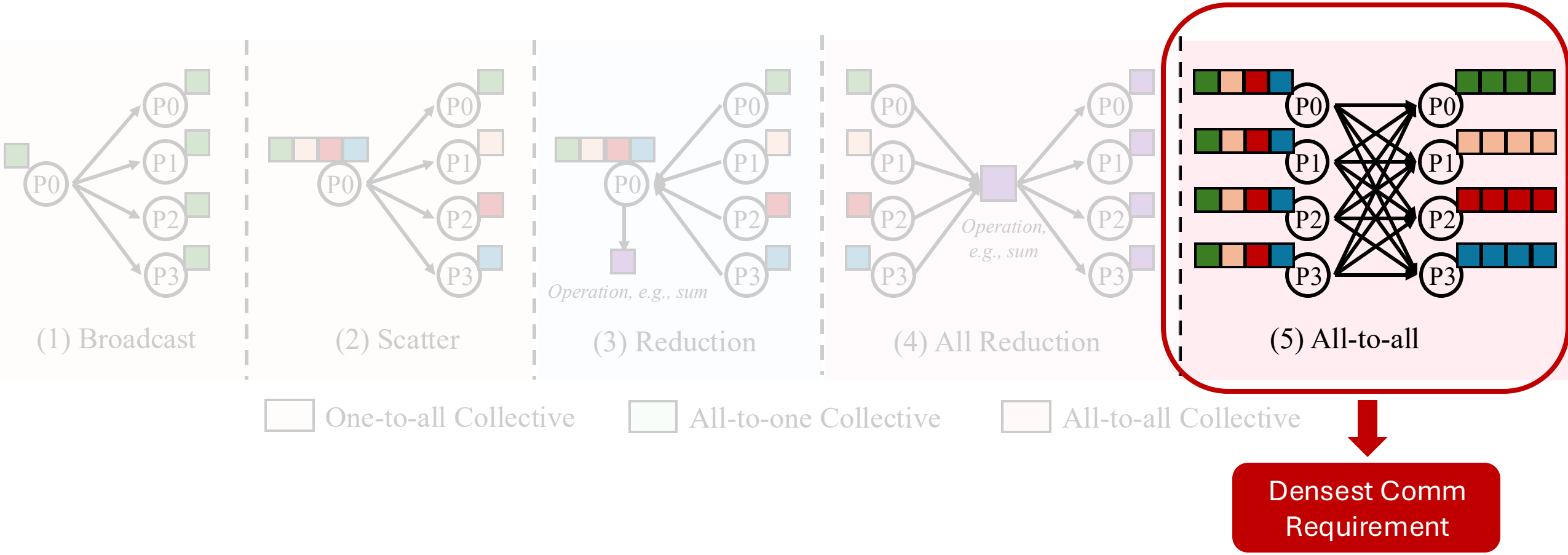


e.g., MPI_Alltoall

Some Widely Used Collective Primitives



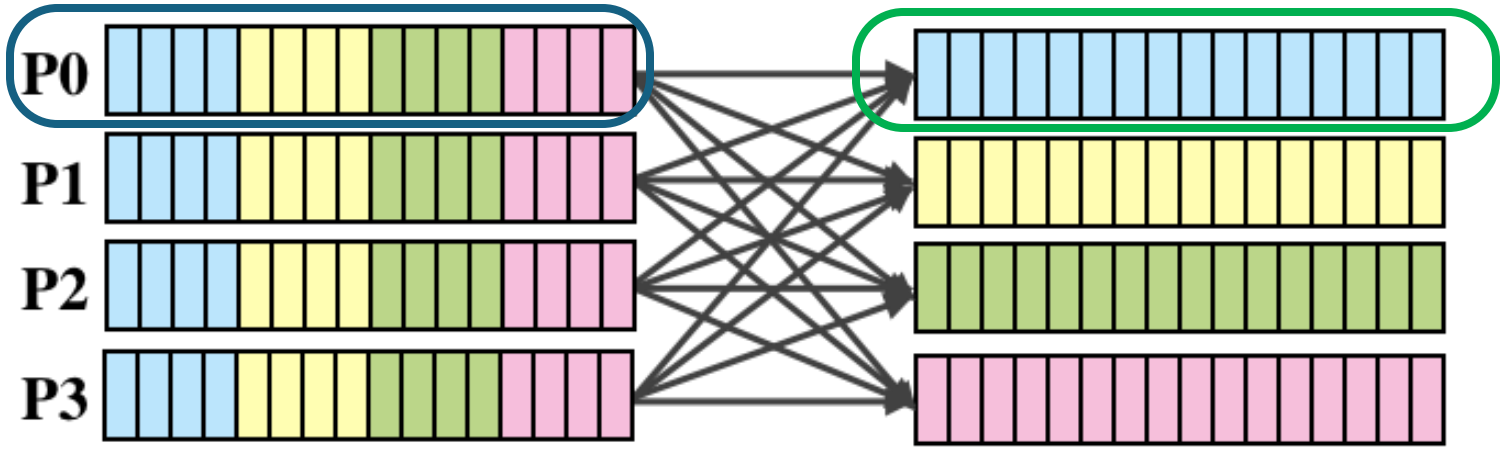
All-to-all Data Exchange is the Most Challenging to Scale and Optimize

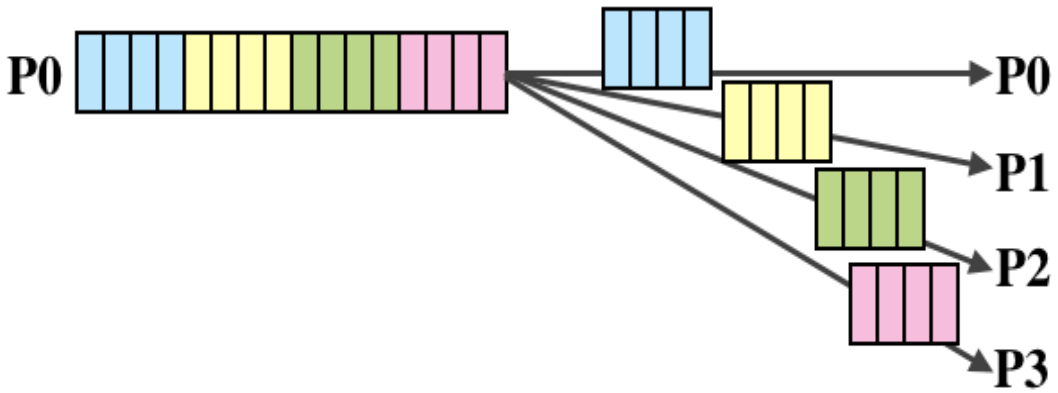


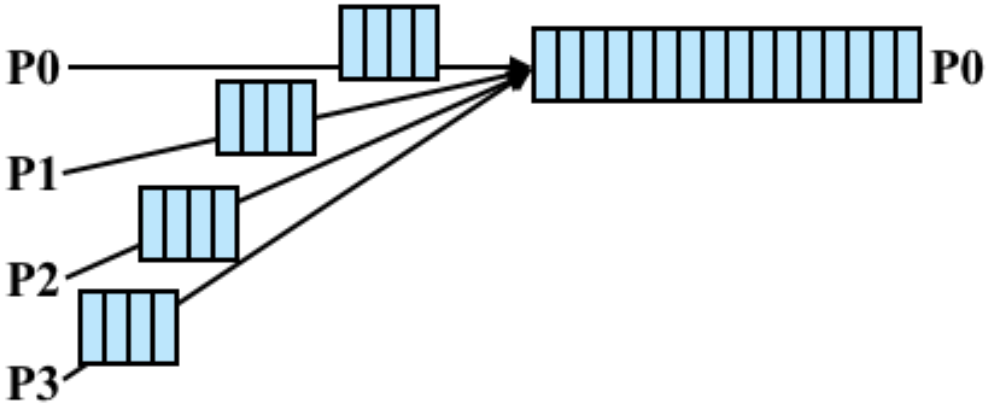
Agenda

- Introduction and Background
  - Inter-process Communication
  - All-to-all Data Exchange
  - Standard MPI All-to-all Implementations
- Challenges and Solutions
  - Tunability of Performance
  - Logarithmic Non-uniform All-to-all
- Parameterized Logarithmic Algorithm
- Parameterized Linear Algorithm
- Evaluation
- Application
- Conclusion

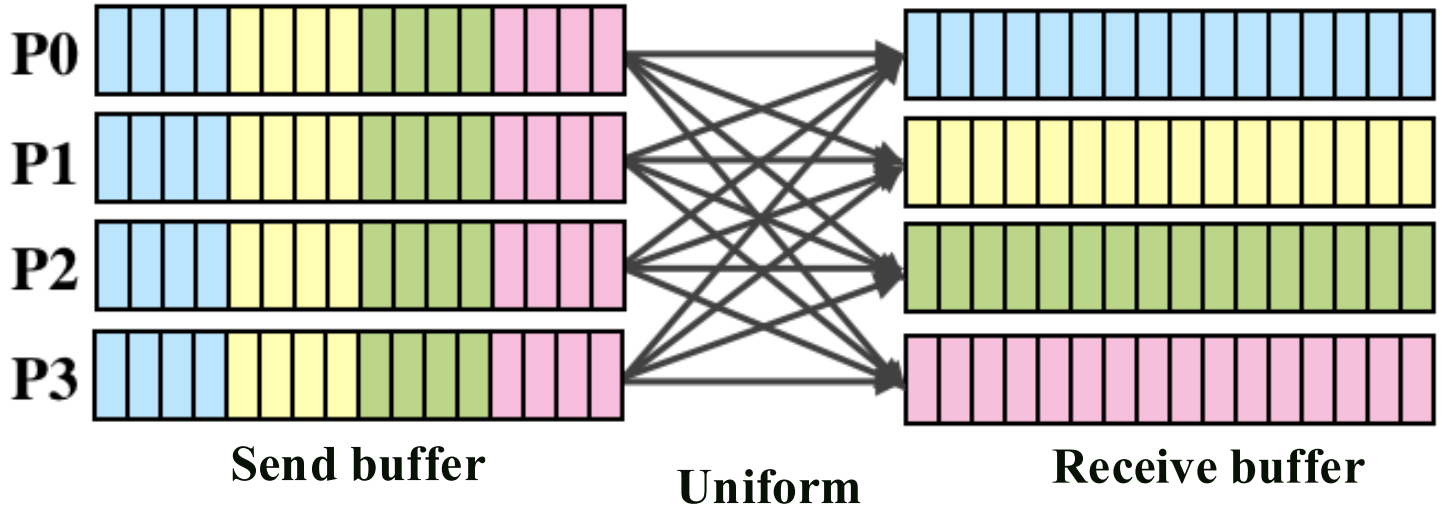
All-to-all – Every process Sends and receives data from every other process

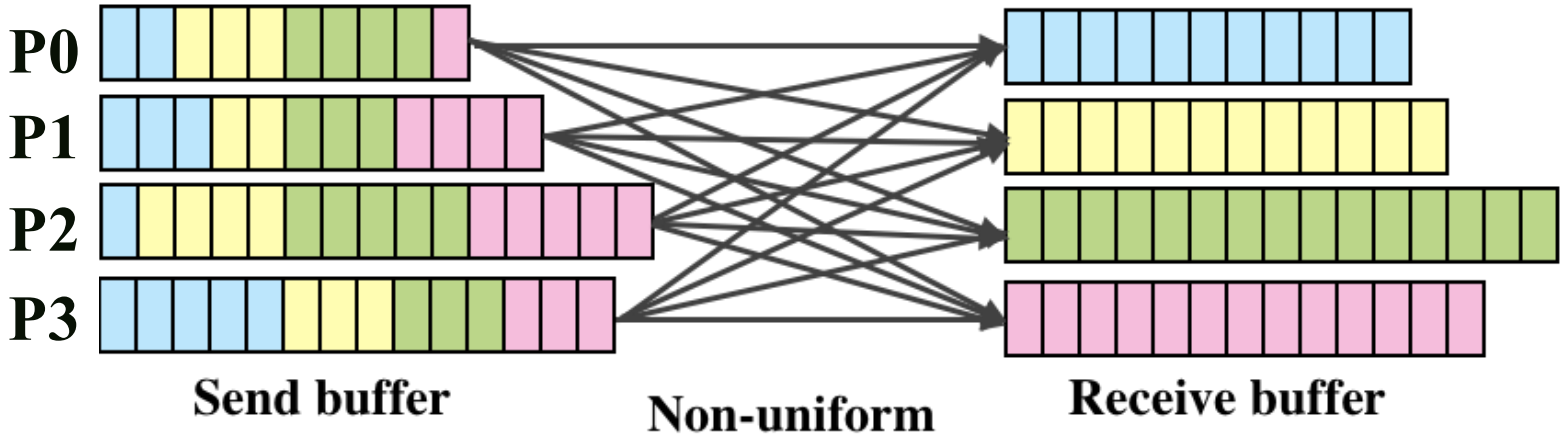






Uniform and Non-uniform All-to-all



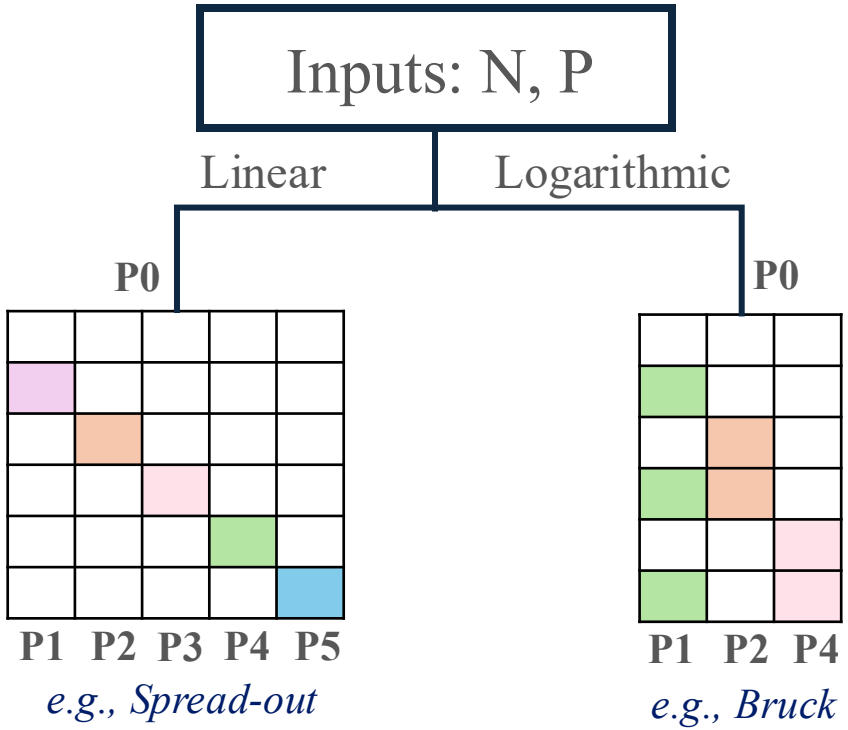


Agenda

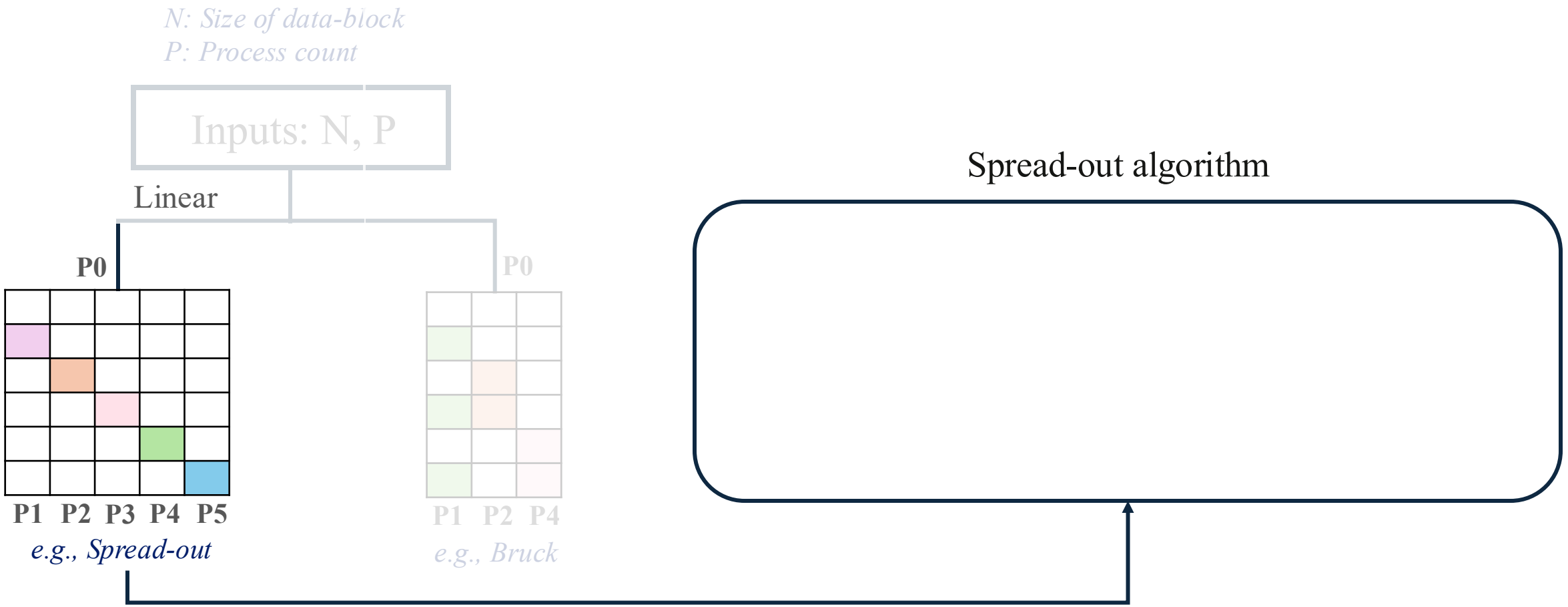
- Introduction and Background
  - Inter-process Communication
  - All-to-all Data Exchange
  - Standard MPI All-to-all Implementations
- Challenges and Solutions
  - Tunability of Performance
  - Logarithmic Non-uniform All-to-all
- Parameterized Logarithmic Algorithm
- Parameterized Linear Algorithm
- Evaluation
- Application
- Conclusion

Standard MPI All-to-all Implementations

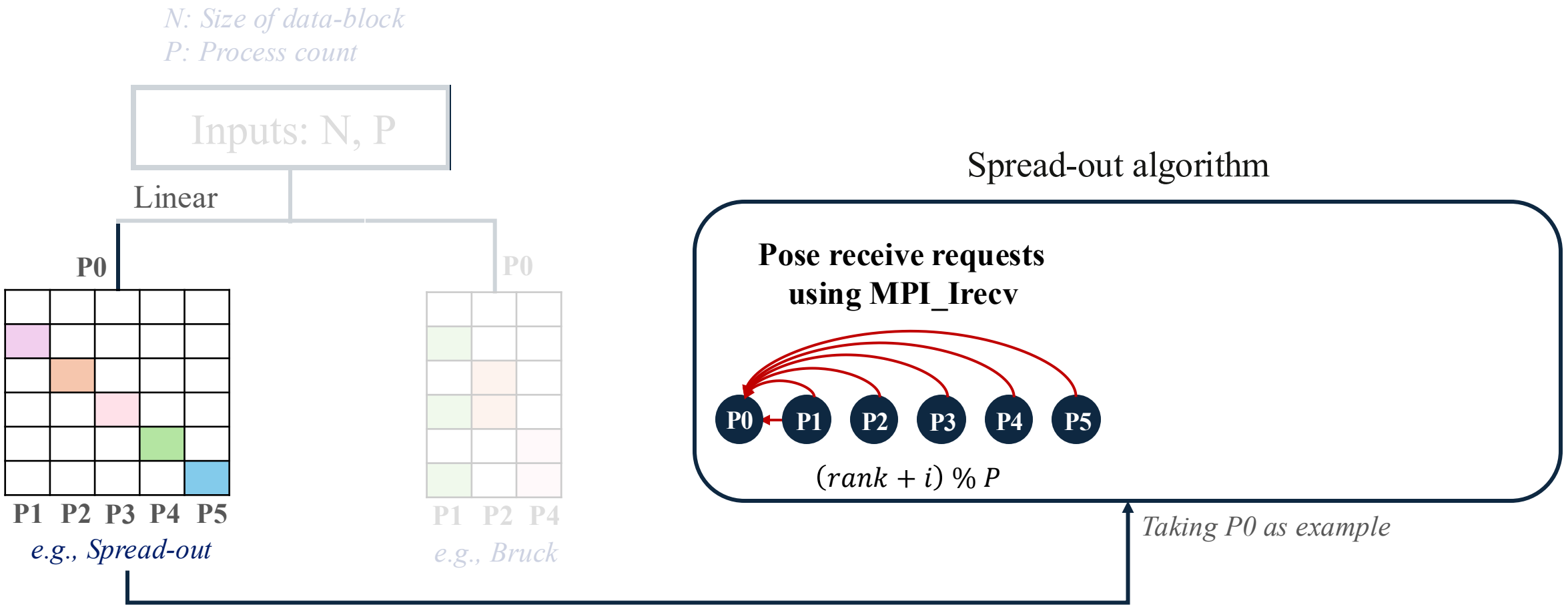
N: Size of data-block
P: Process count



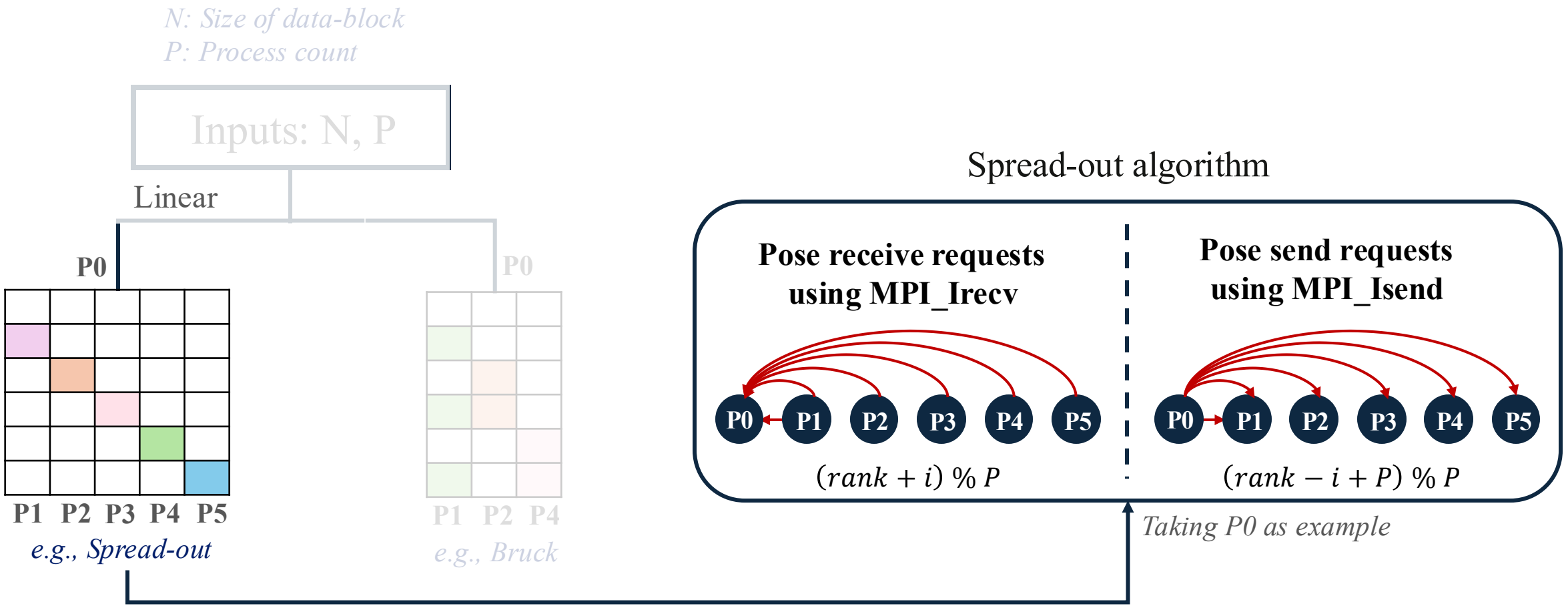
Standard MPI All-to-all Implementations: Spread-out



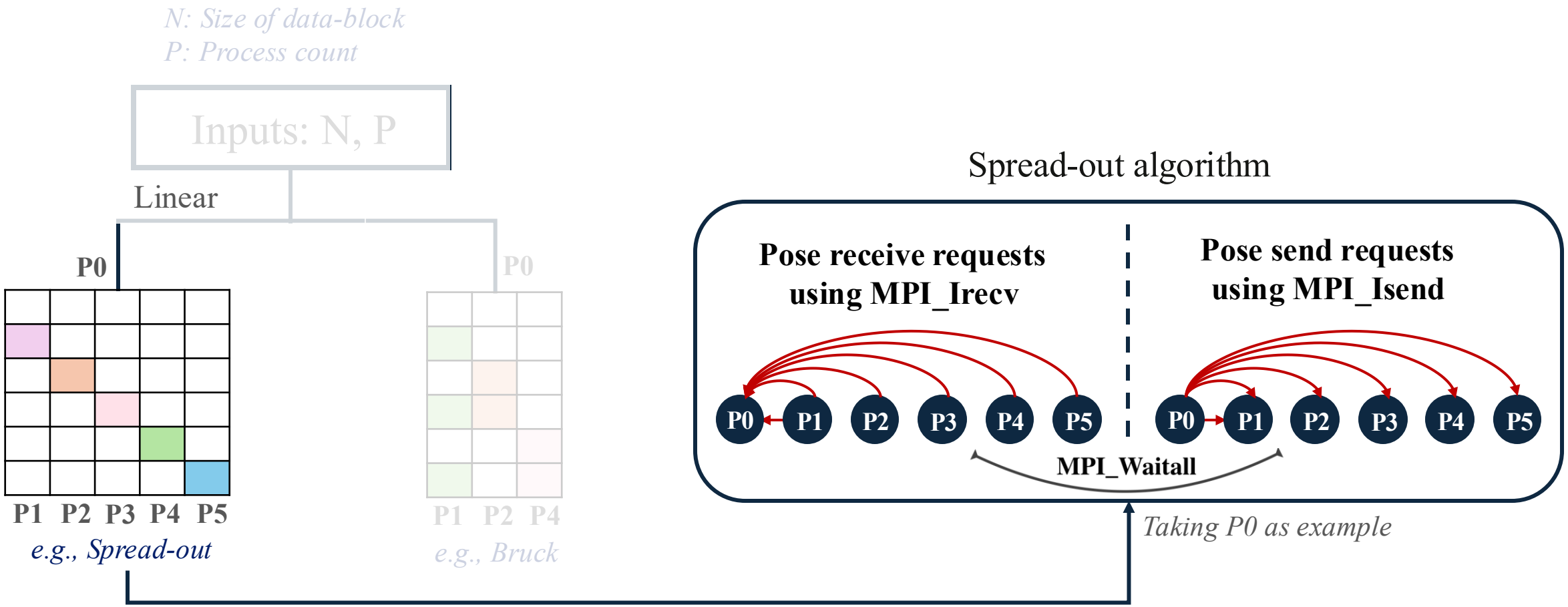
Standard MPI All-to-all Implementations: Spread-out



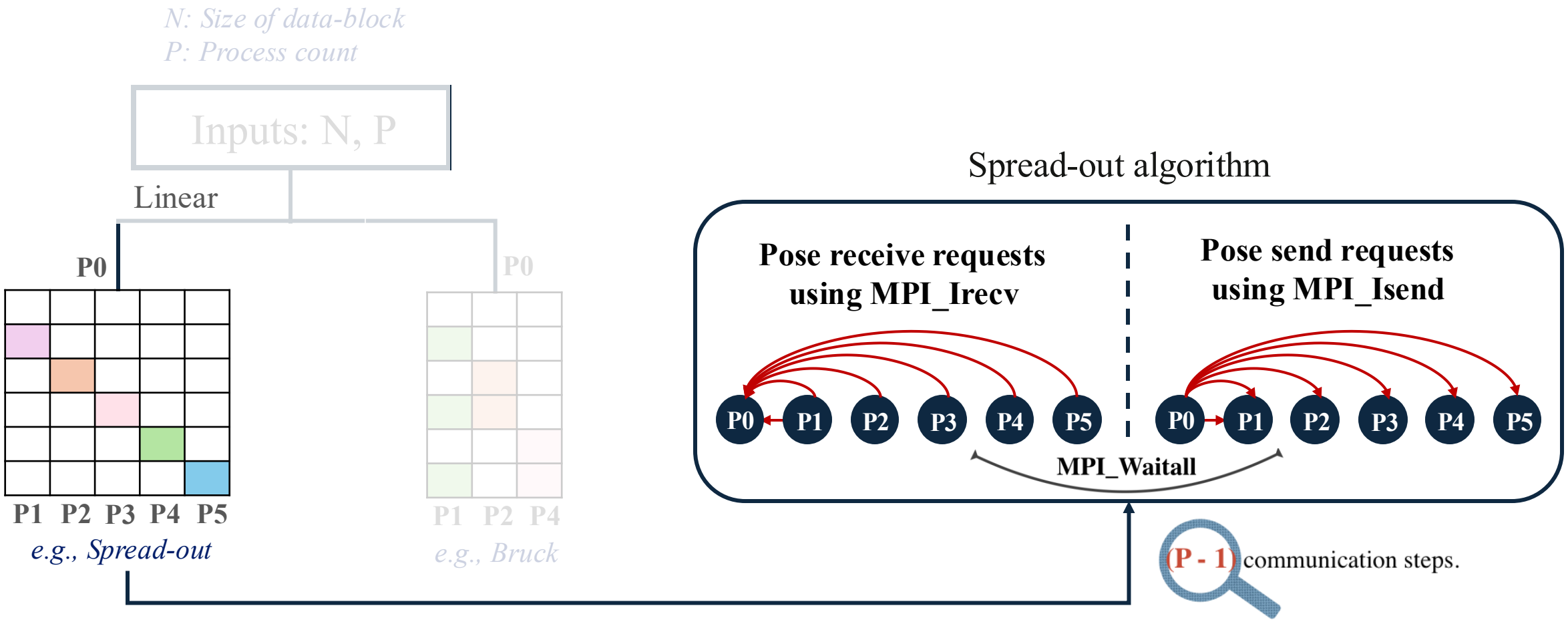
Standard MPI All-to-all Implementations: Spread-out



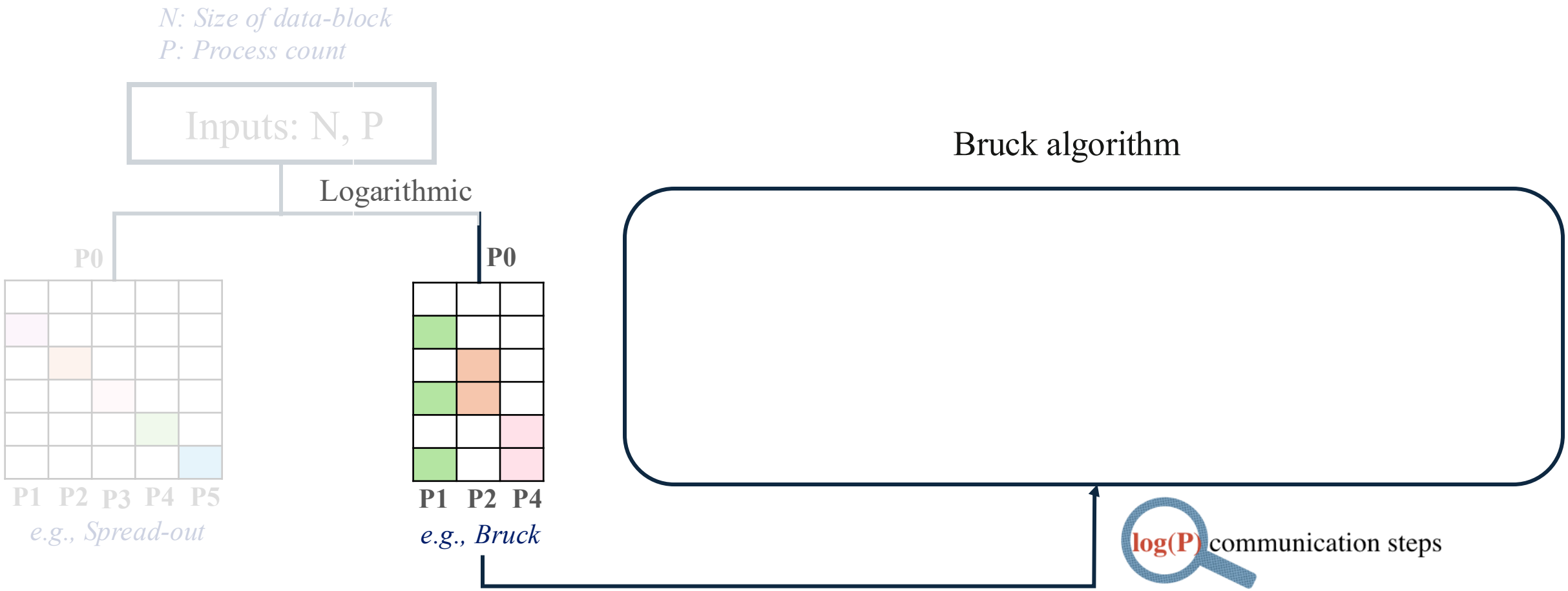
Standard MPI All-to-all Implementations: Spread-out



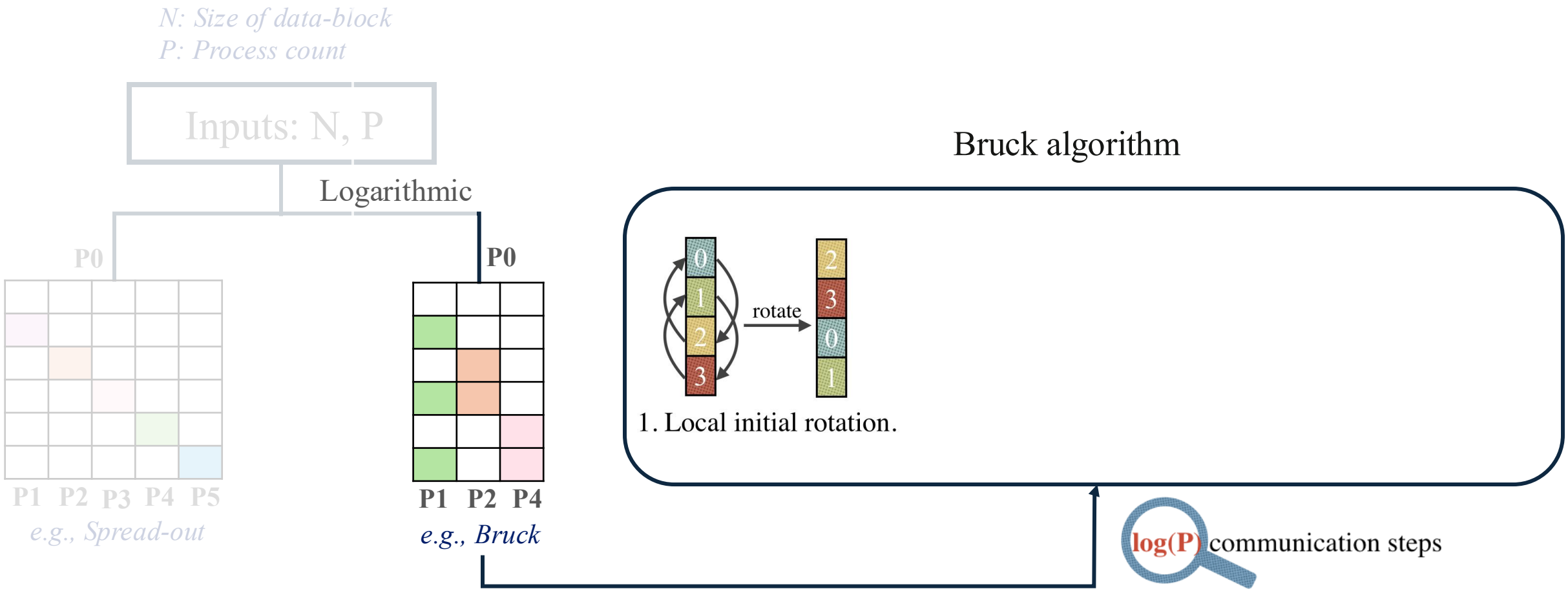
Standard MPI All-to-all Implementations: Spread-out



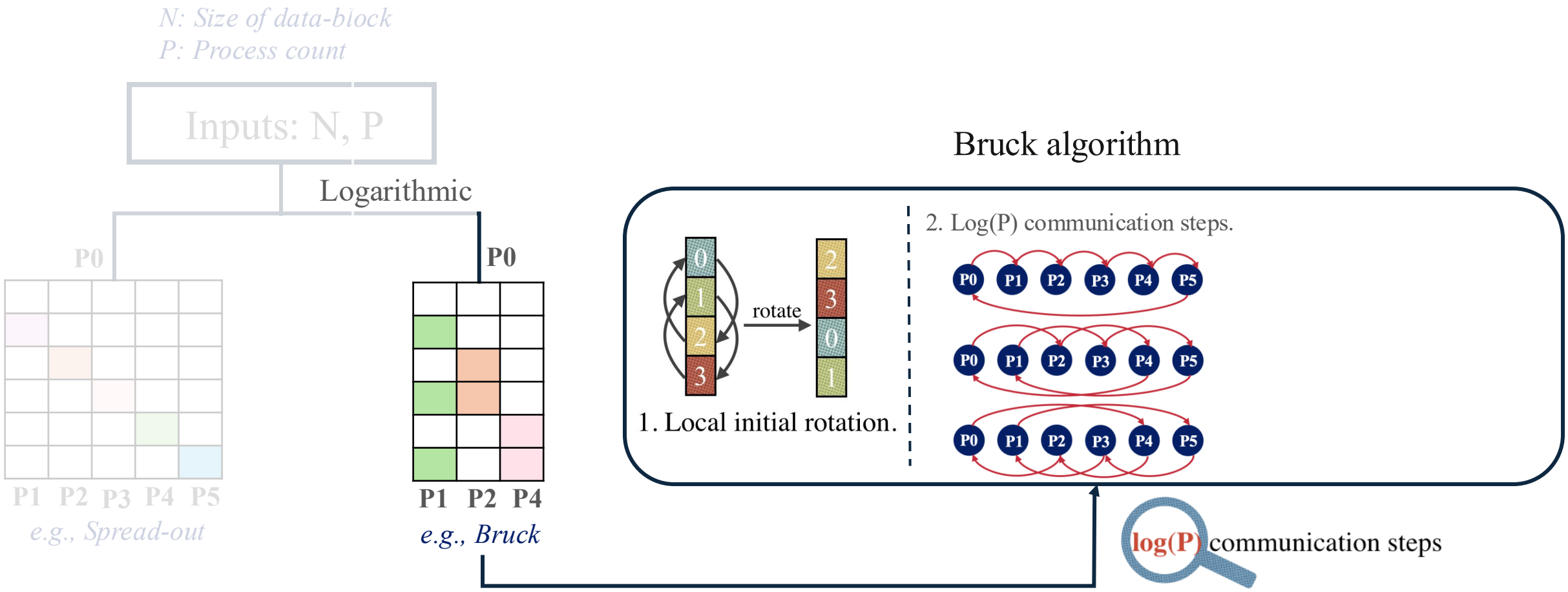
Standard MPI All-to-all Implementations: Bruck



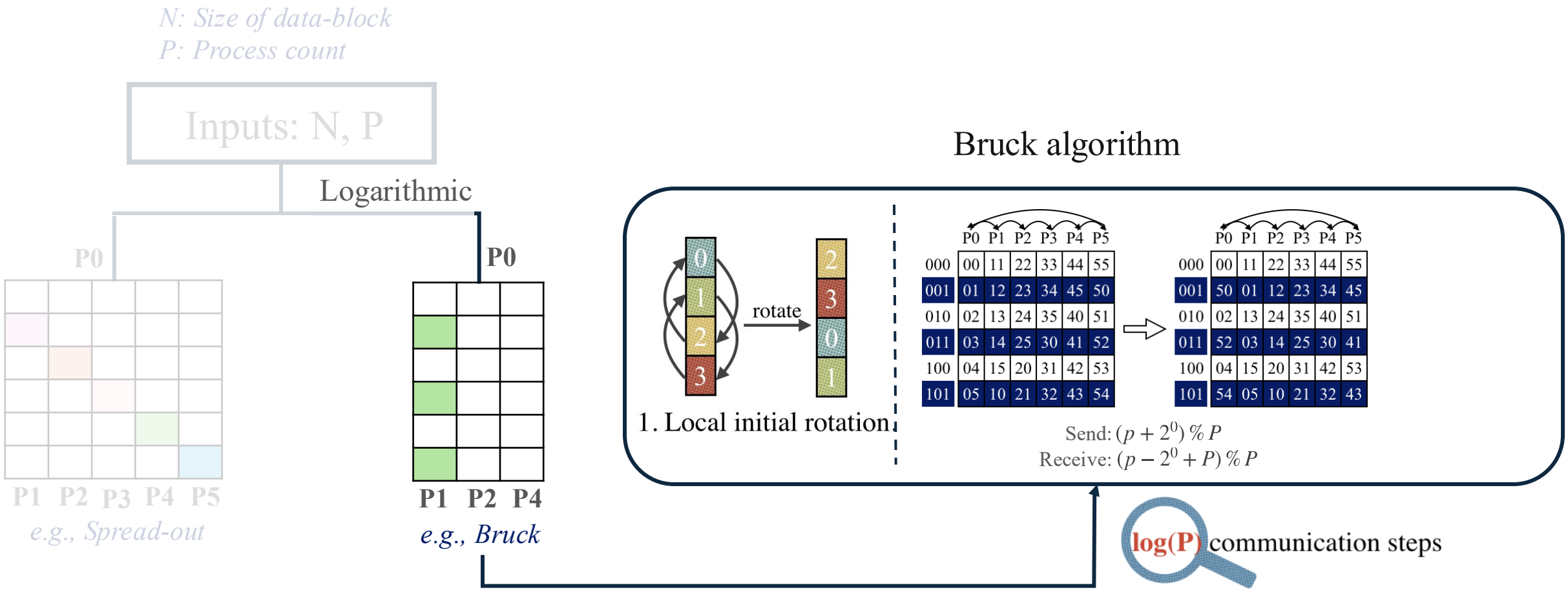
Standard MPI All-to-all Implementations: Bruck



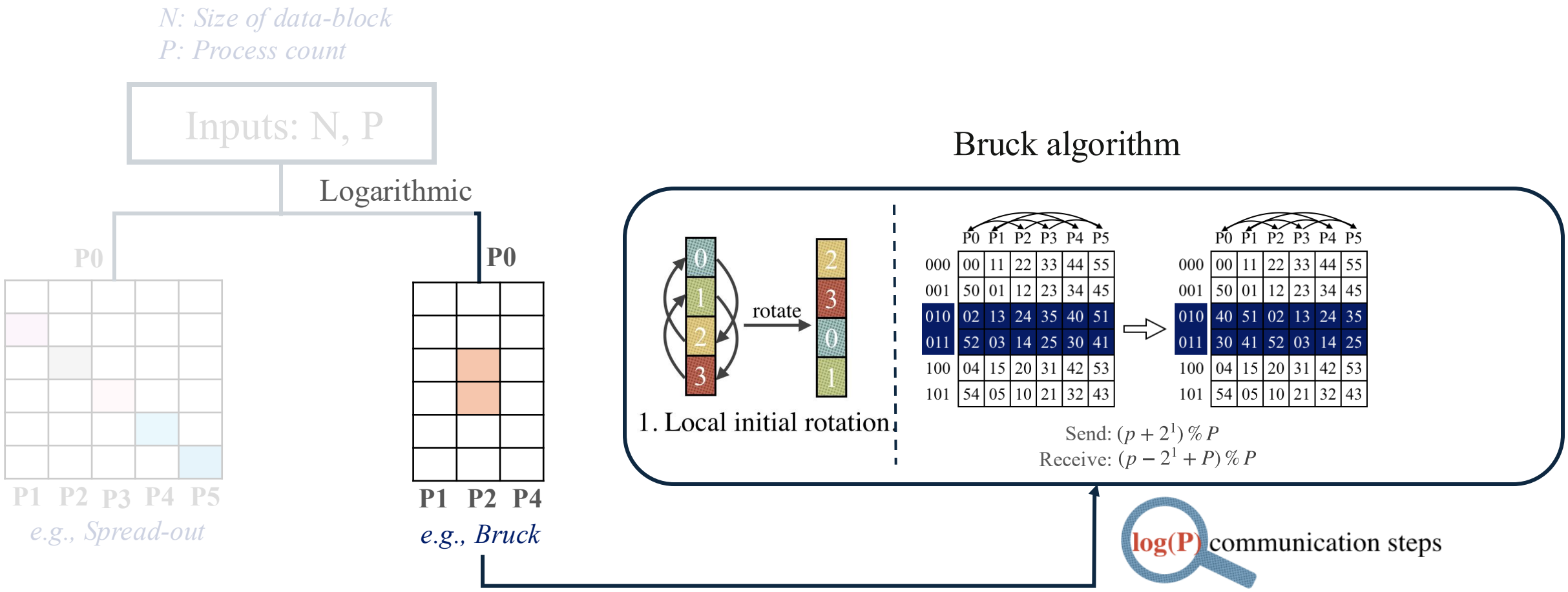
Standard MPI All-to-all Implementations: Bruck



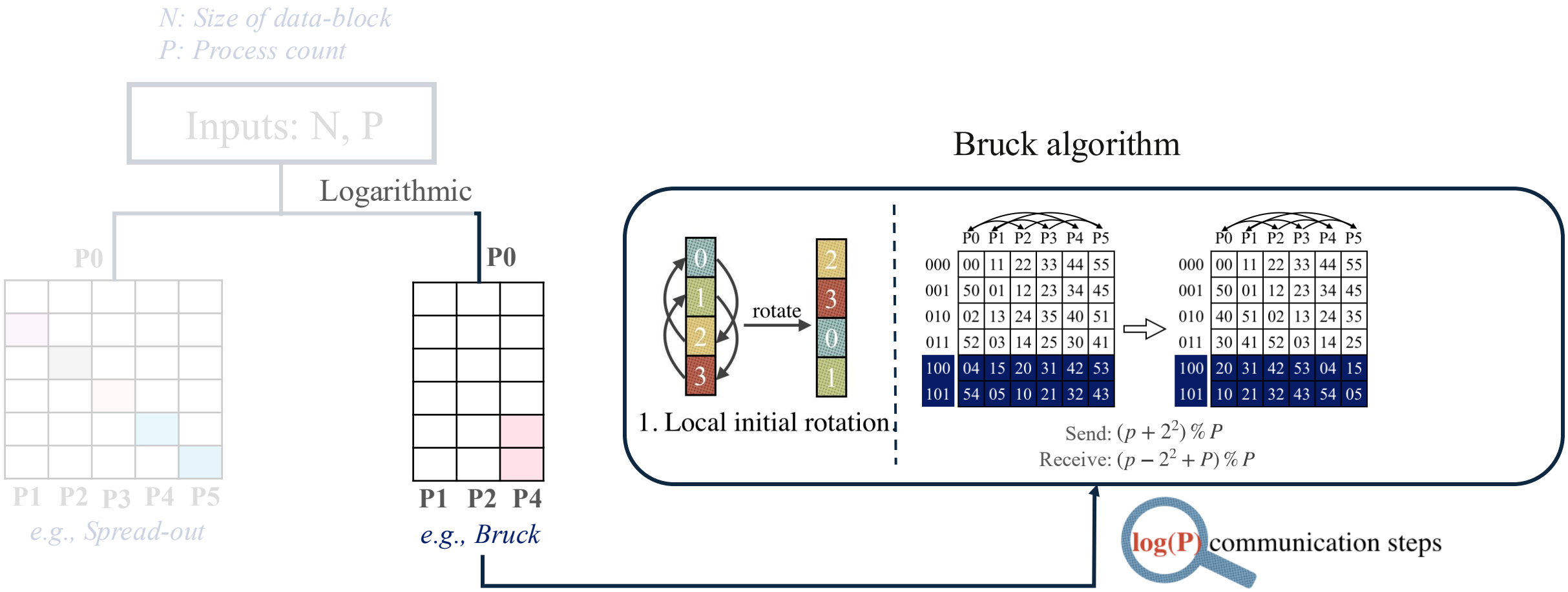
Standard MPI All-to-all Implementations: Comm-0



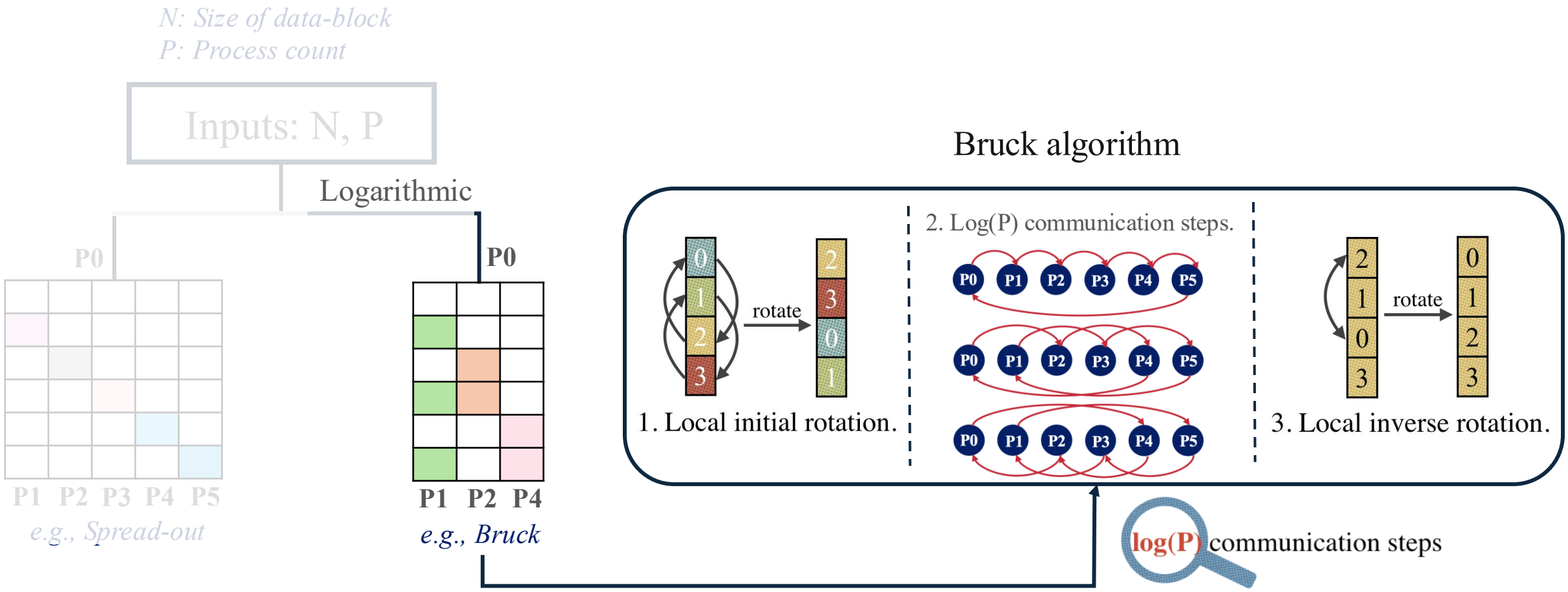
Standard MPI All-to-all Implementations: Comm-1



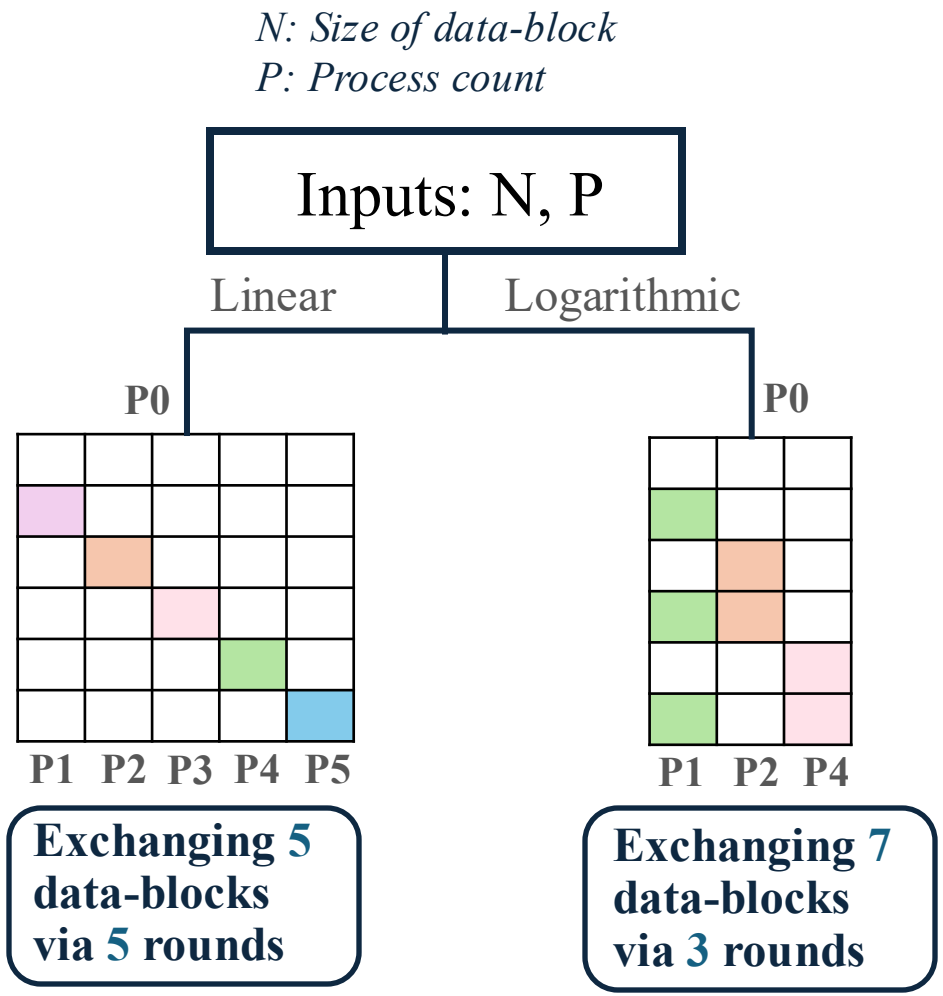
Standard MPI All-to-all Implementations: Comm-2



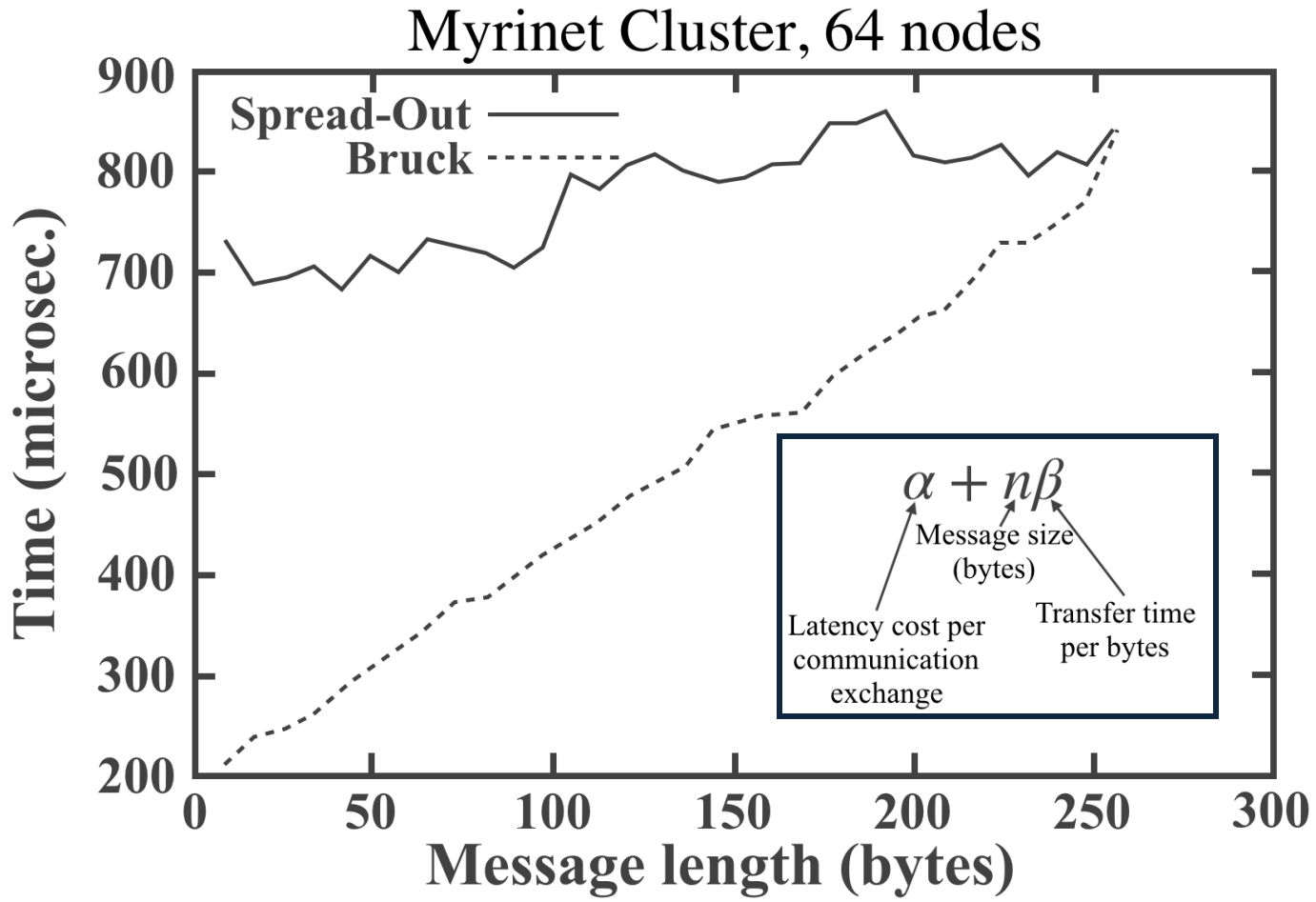
Standard MPI All-to-all Implementations: Bruck



Comparison of These Two Algorithms



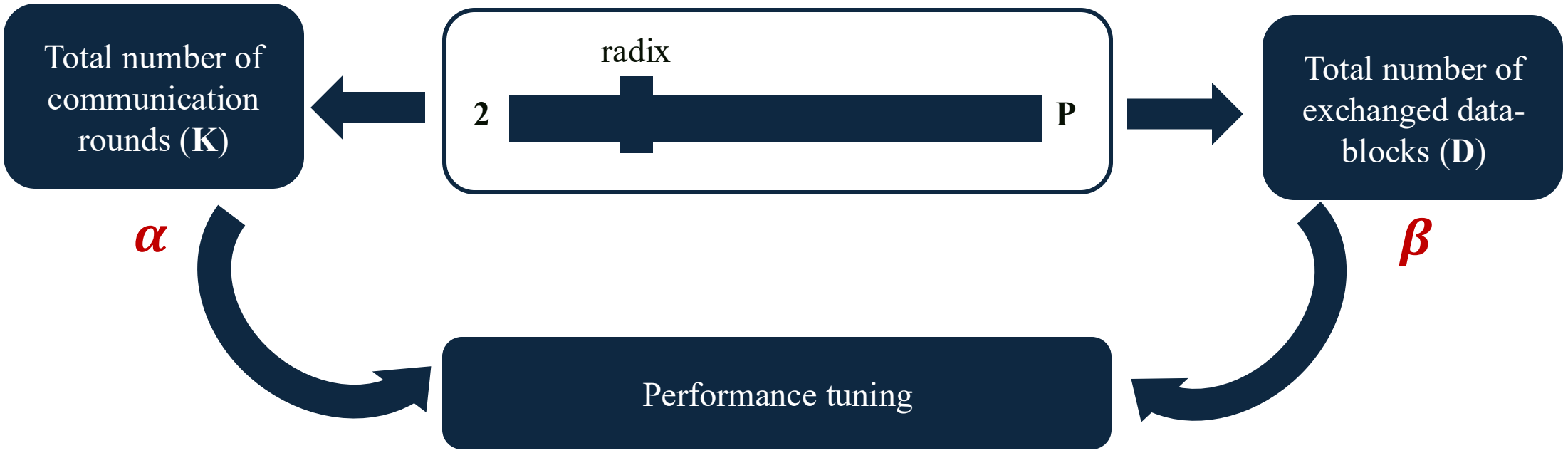
Comparison of These Two Algorithms



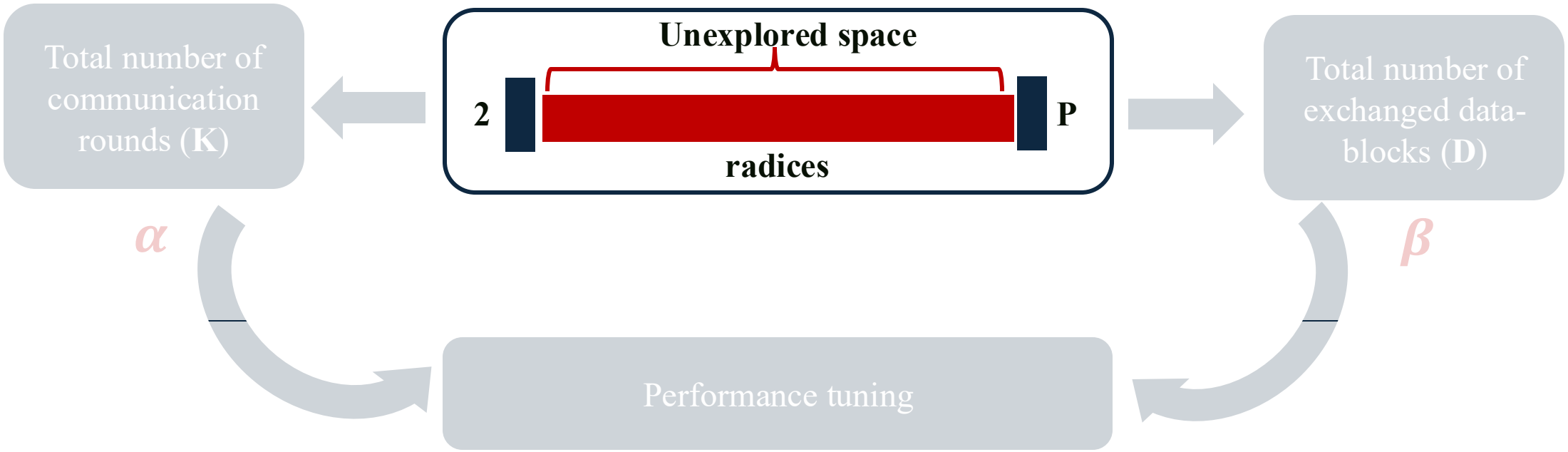
Agenda

- Introduction and Background
  - Inter-process Communication
  - All-to-all Data Exchange
  - Standard MPI All-to-all Implementations
- Challenges and Solutions
  - Tunability of Performance
  - Logarithmic Non-uniform All-to-all
- Parameterized Logarithmic Algorithm
- Parameterized Linear Algorithm
- Evaluation
- Application
- Conclusion

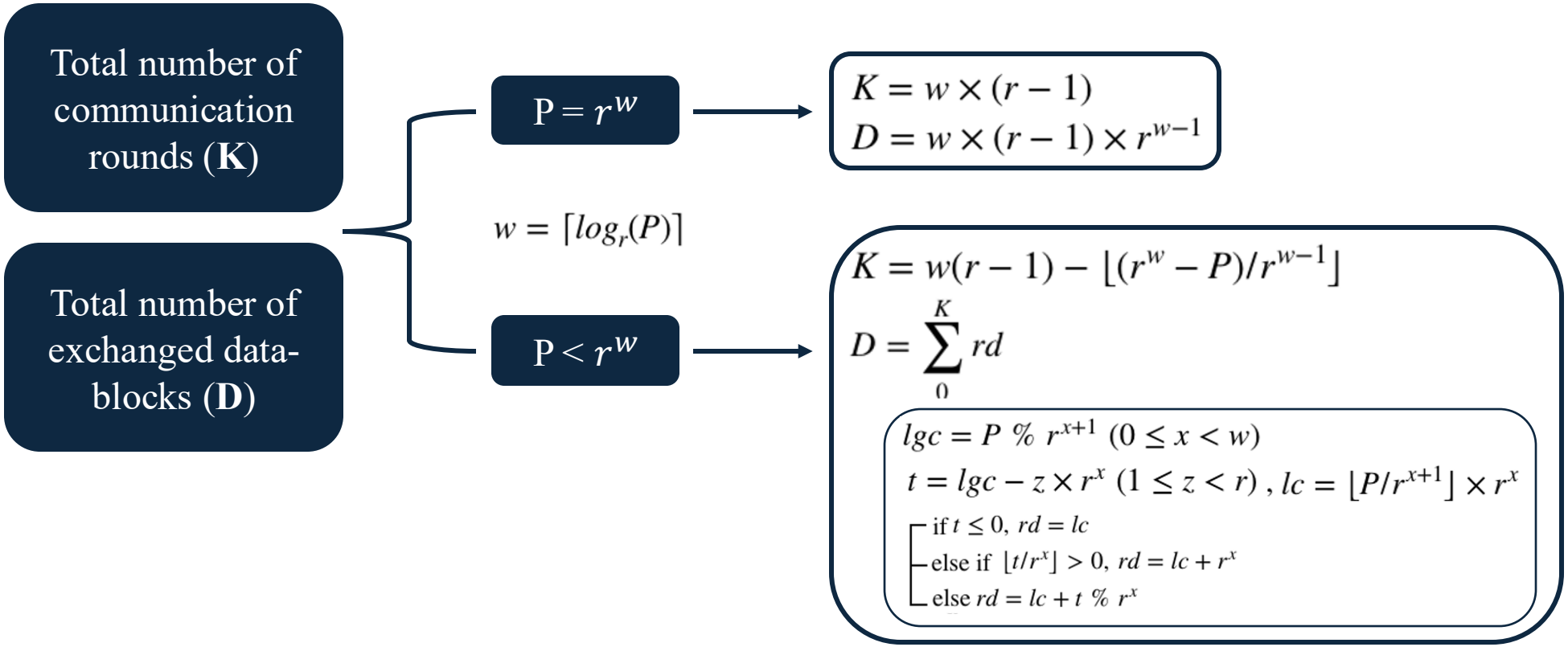
Tuning Radices Enables Performance Tuning



Standard MPI All-to-all Implementation Leaves a Large Unexplored Space

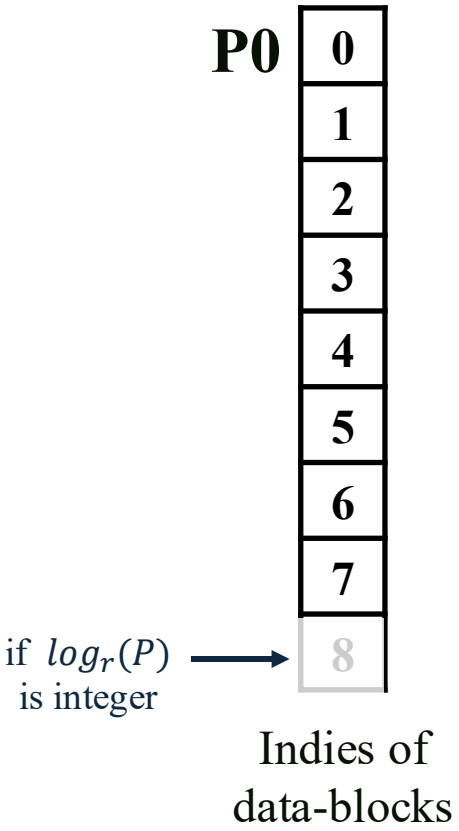


Mathematics of Tunable Radix All-to-all

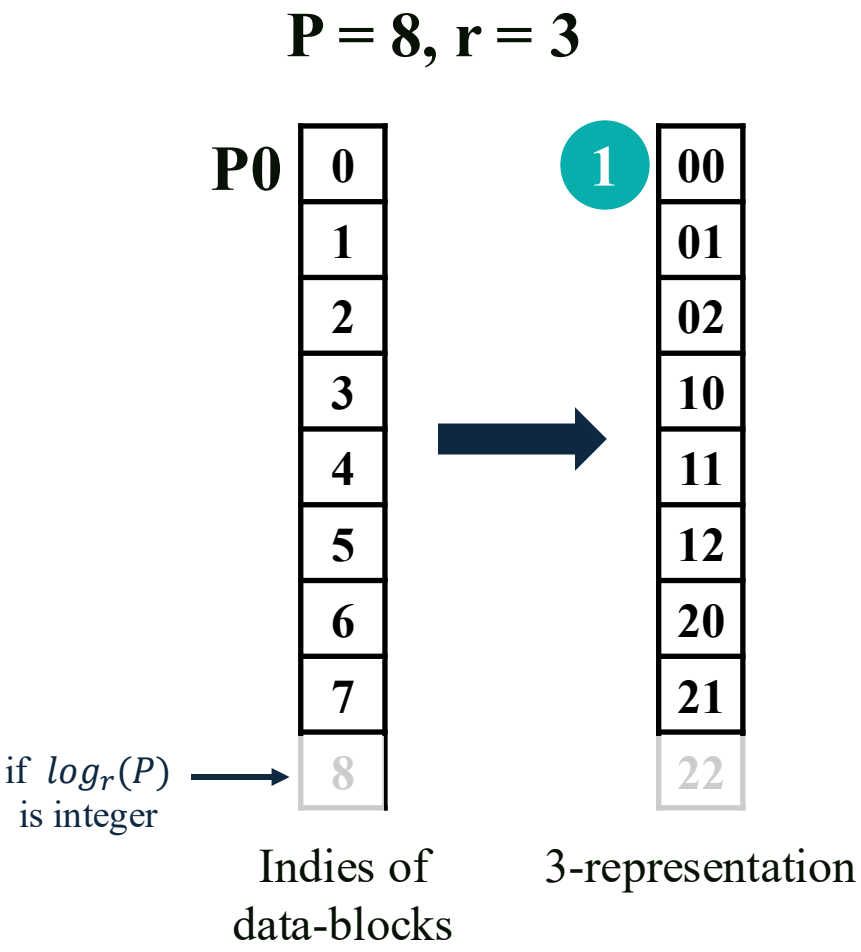


Tunable Radix All-to-all Example

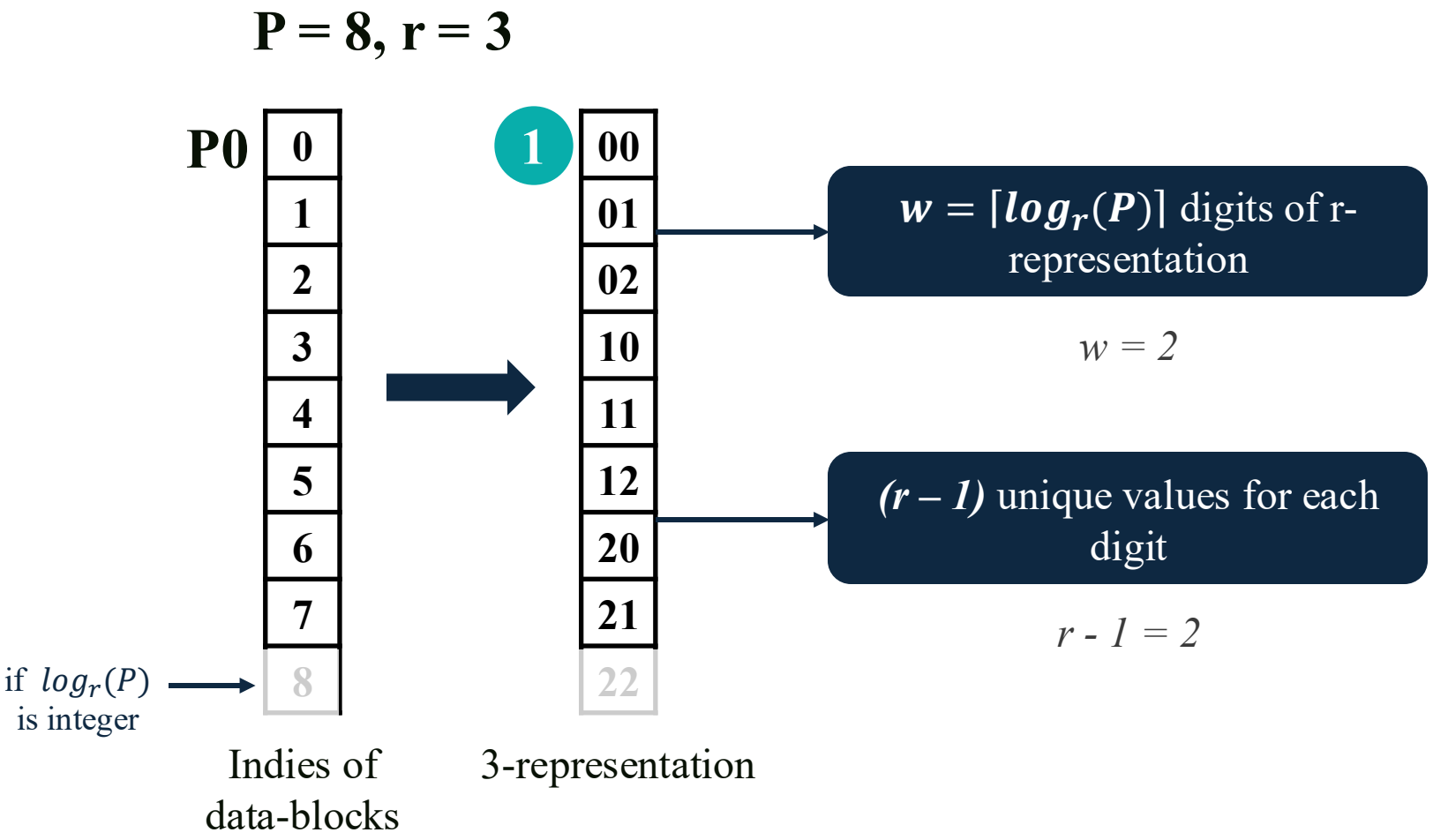
P = 8, r = 3



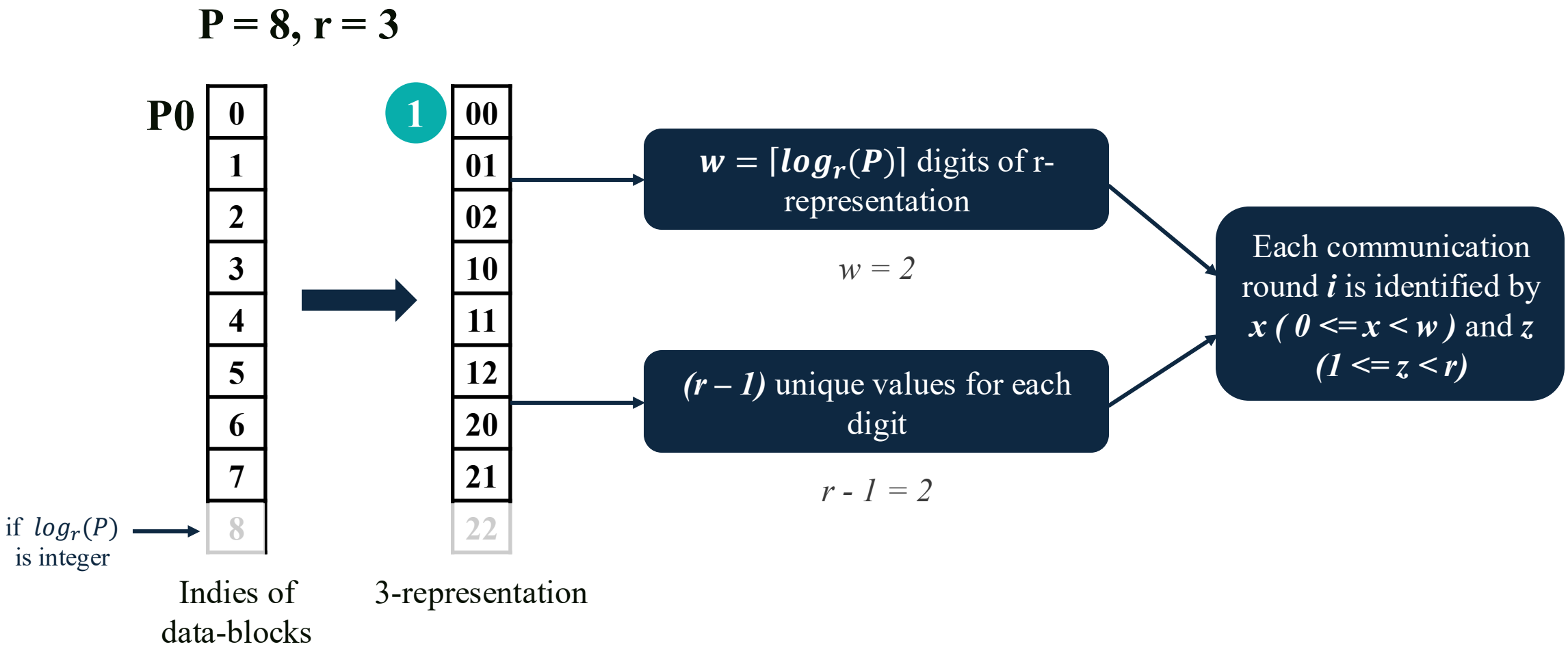
Tunable Radix All-to-all Example: r-representation



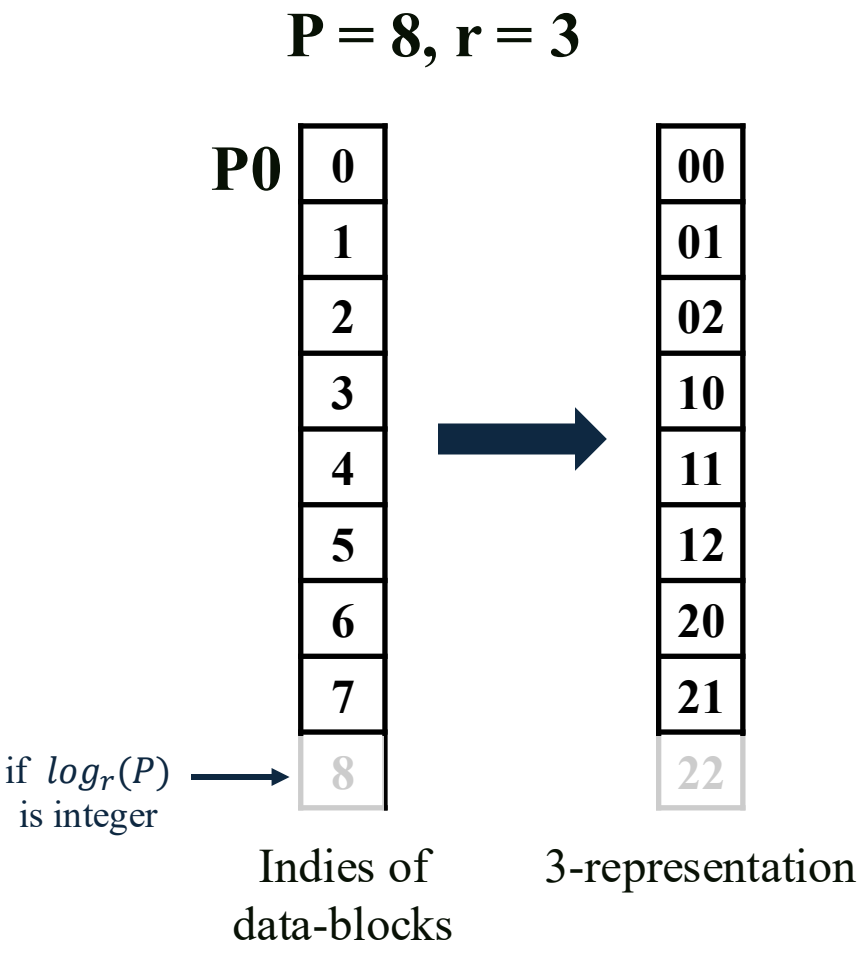
Tunable Radix All-to-all Example: r-representation



Tunable Radix All-to-all Example: r-representation

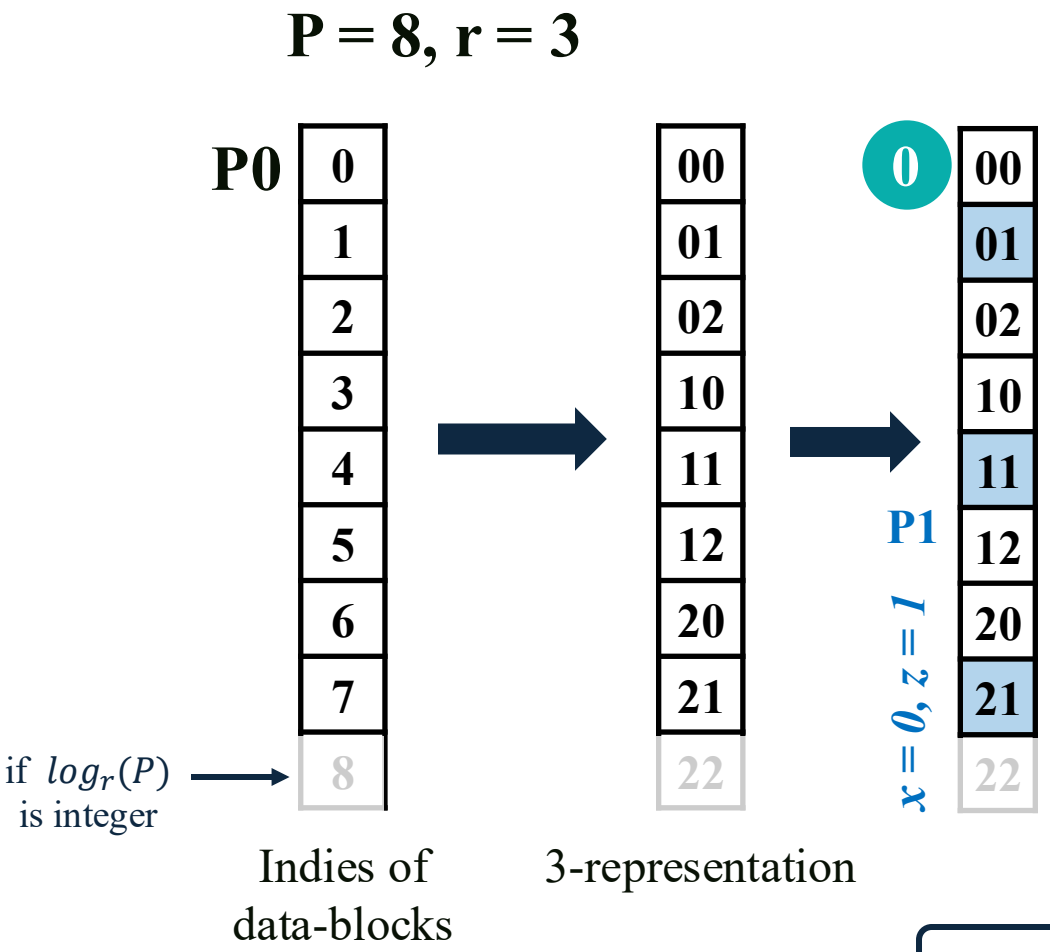


Tunable Radix All-to-all Example: Communication



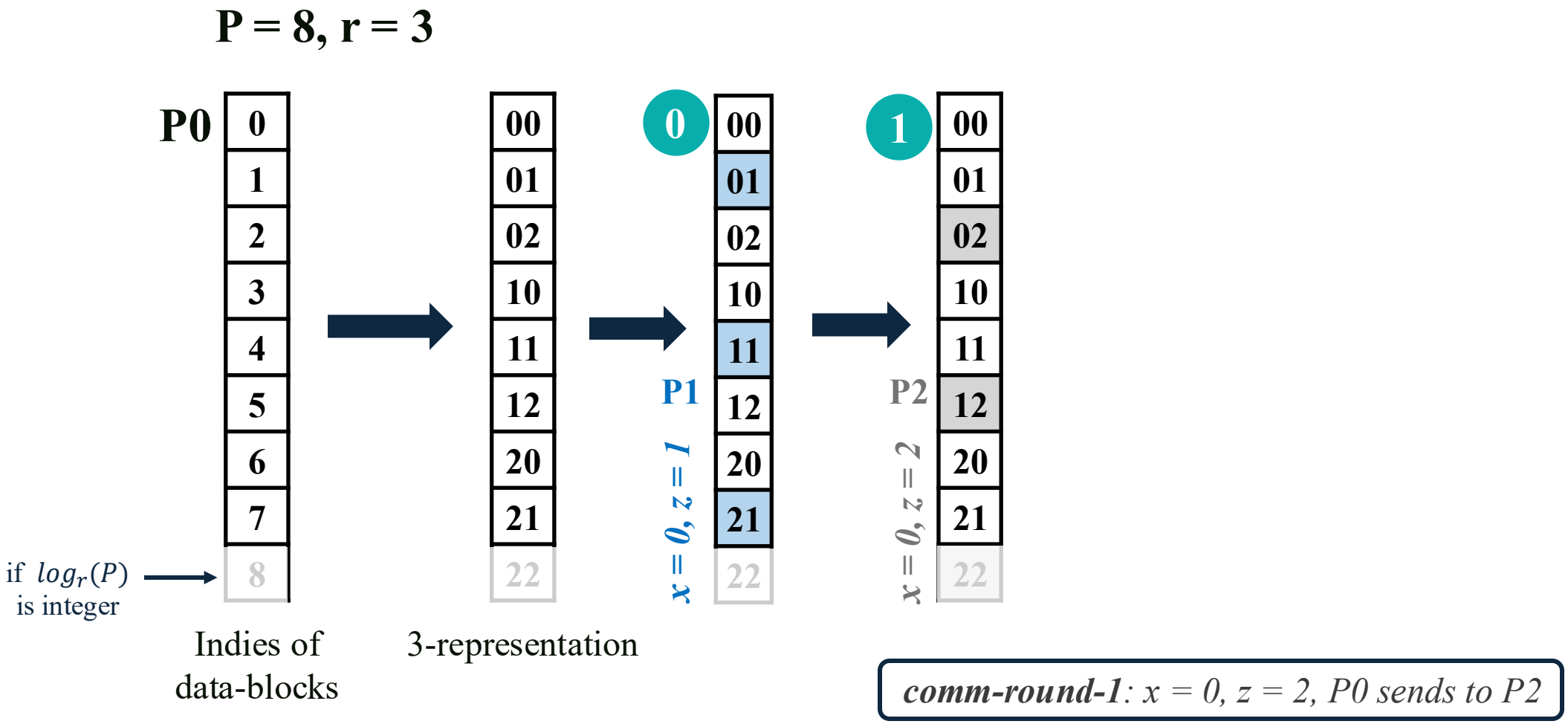
P0 sends data-blocks whose **x** digit equals to z

Tunable Radix All-to-all Example: Comm-round-0

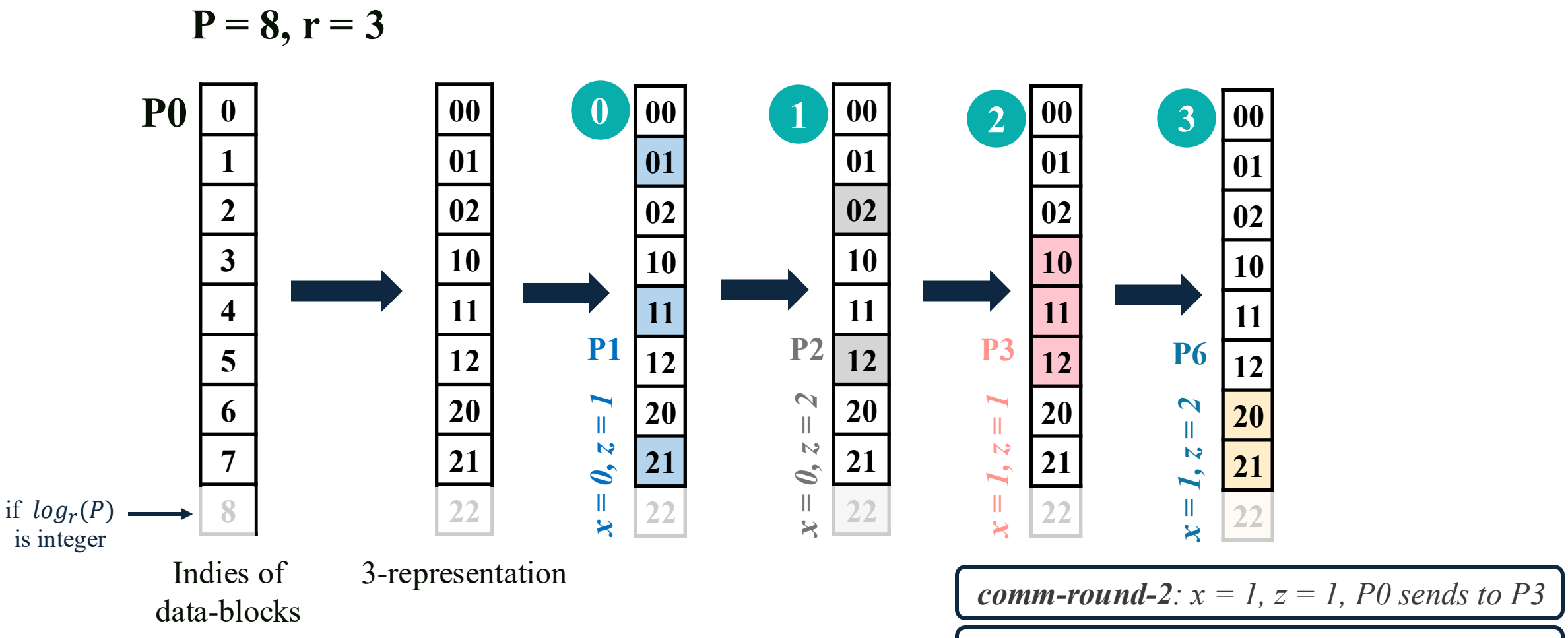


***comm-round-0**: x = 0, z = 1, P0 sends to P1*

Tunable Radix All-to-all Example: Comm-round-1



Tunable Radix All-to-all Example: Comm-round-2,3



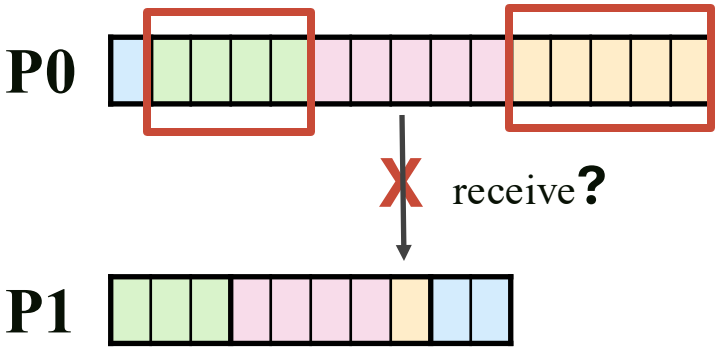
comm-round-2: *x = 1, z = 1, P0 sends to P3*

comm-round-3: *x = 1, z = 2, P0 sends to P6*

Agenda

- Introduction and Background
  - Inter-process Communication
  - All-to-all Data Exchange
  - Standard MPI All-to-all Implementations
- Challenges and Solutions
  - Tunability of Performance
  - Logarithmic Non-uniform All-to-all
- Parameterized Logarithmic Algorithm
- Parameterized Linear Algorithm
- Evaluation
- Application
- Conclusion

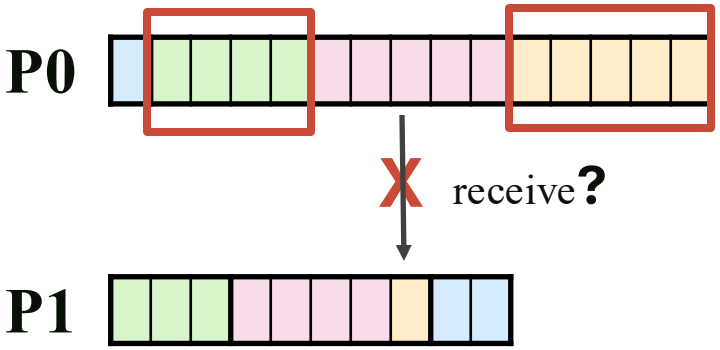
Challenges of Applying the Logarithmic Algorithm to Non-uniform All-to-all Data Exchange



Each process is unaware of how much data to expect during each intermediate communication round.

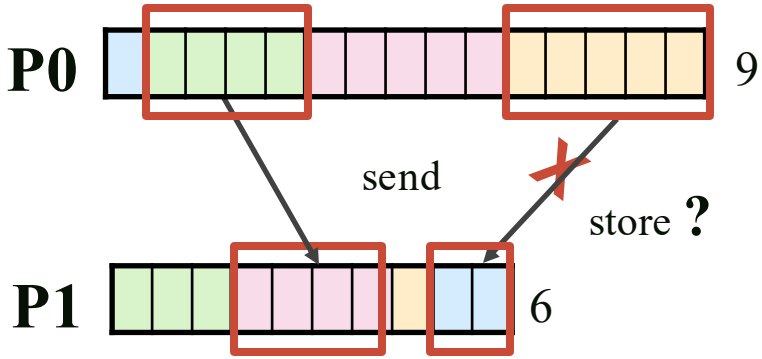
Non-uniform workloads

Challenges of Applying the Logarithmic Algorithm to Non-uniform All-to-all Data Exchange



Each process is unaware of how much data to expect during each intermediate communication round.

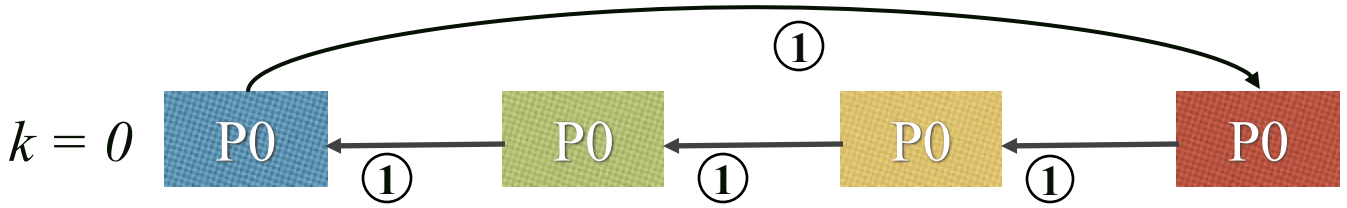
Non-uniform workloads



The received data elements could be too large to fit into the segment in the send buffer.

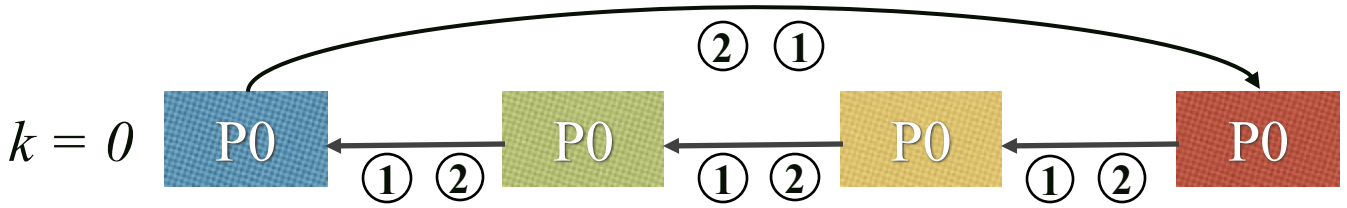
Store-and-forward

Two-phase Communication Scheme



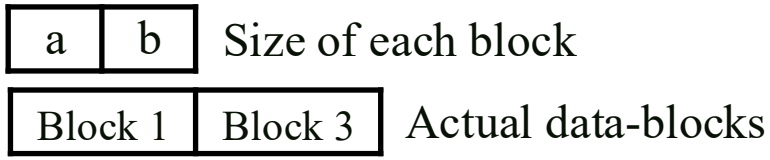
① *Metadata Exchange* | a | b | Size of each block

Two-phase Communication Scheme

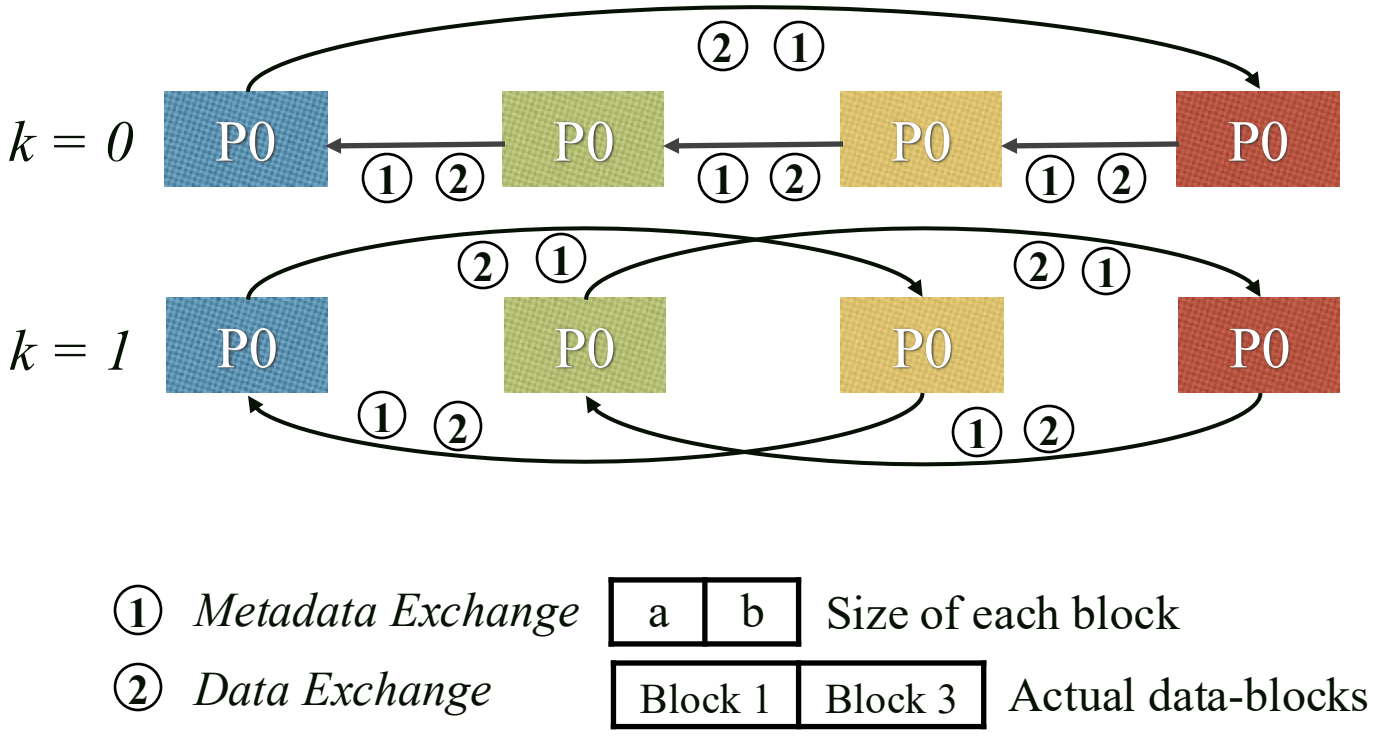


① *Metadata Exchange*

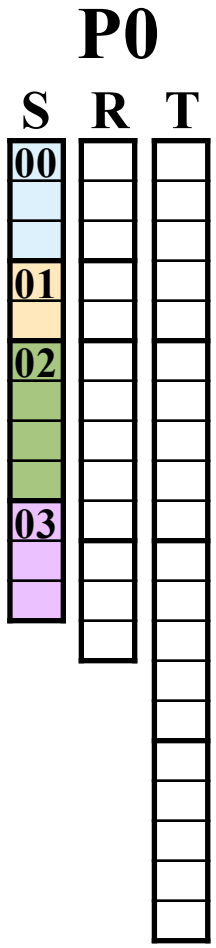
② *Data Exchange*



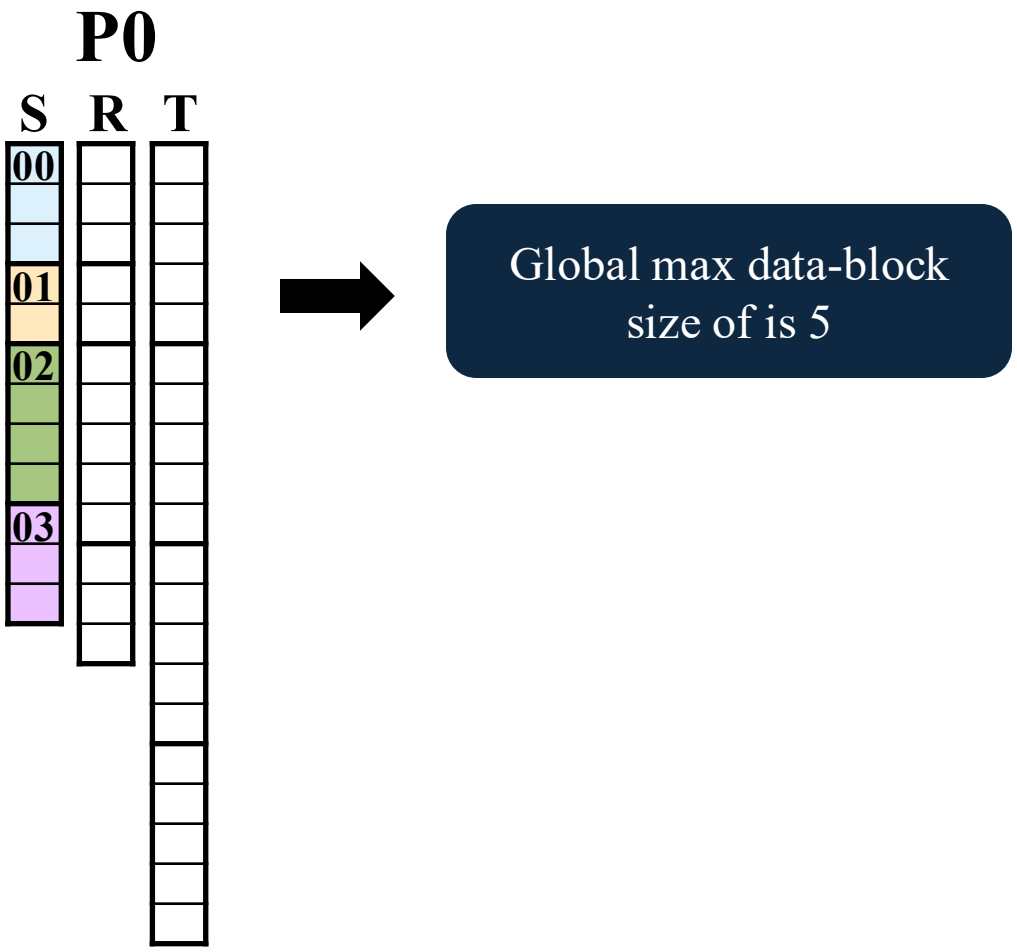
Two-phase Communication Scheme



Using Temporary Buffer to Solve Challenge 2

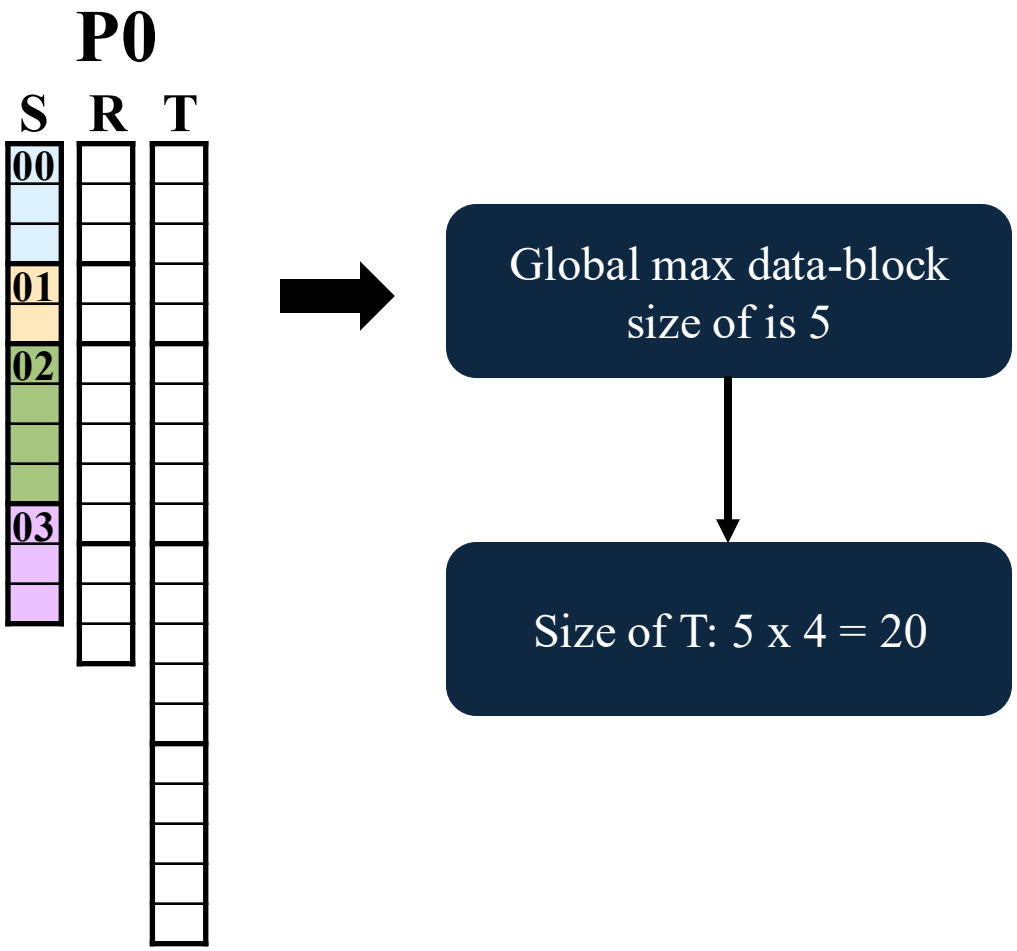


Estimating Temporary Buffer Size

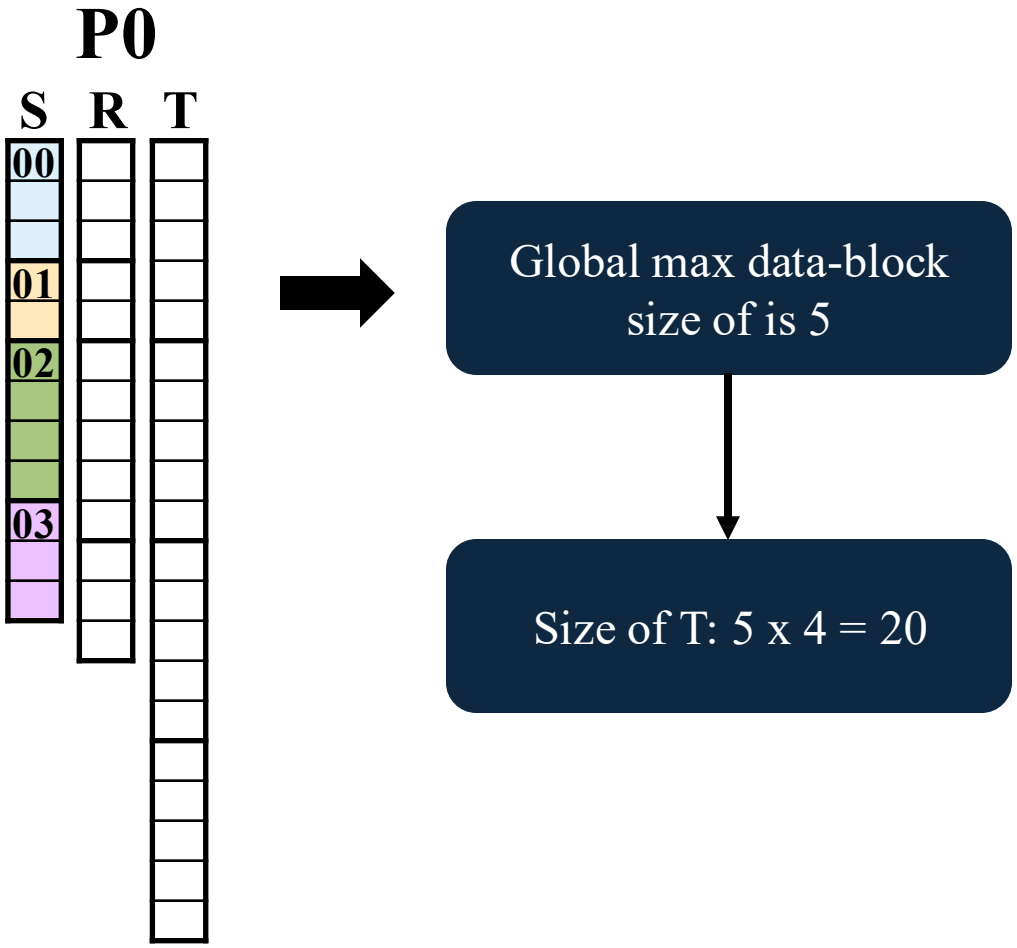


Global max data-block size of is 5

Estimating Temporary Buffer Size



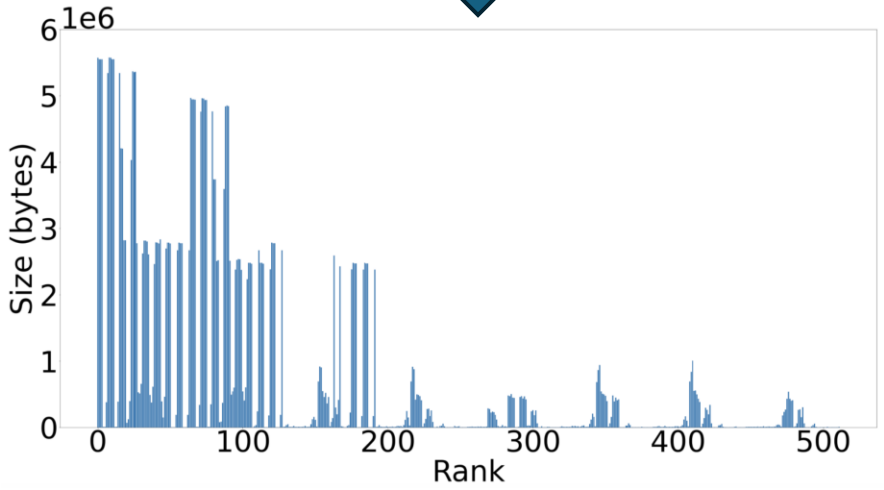
Temporary Buffer Size Can Be Very Large



Global max data-block size of is 5

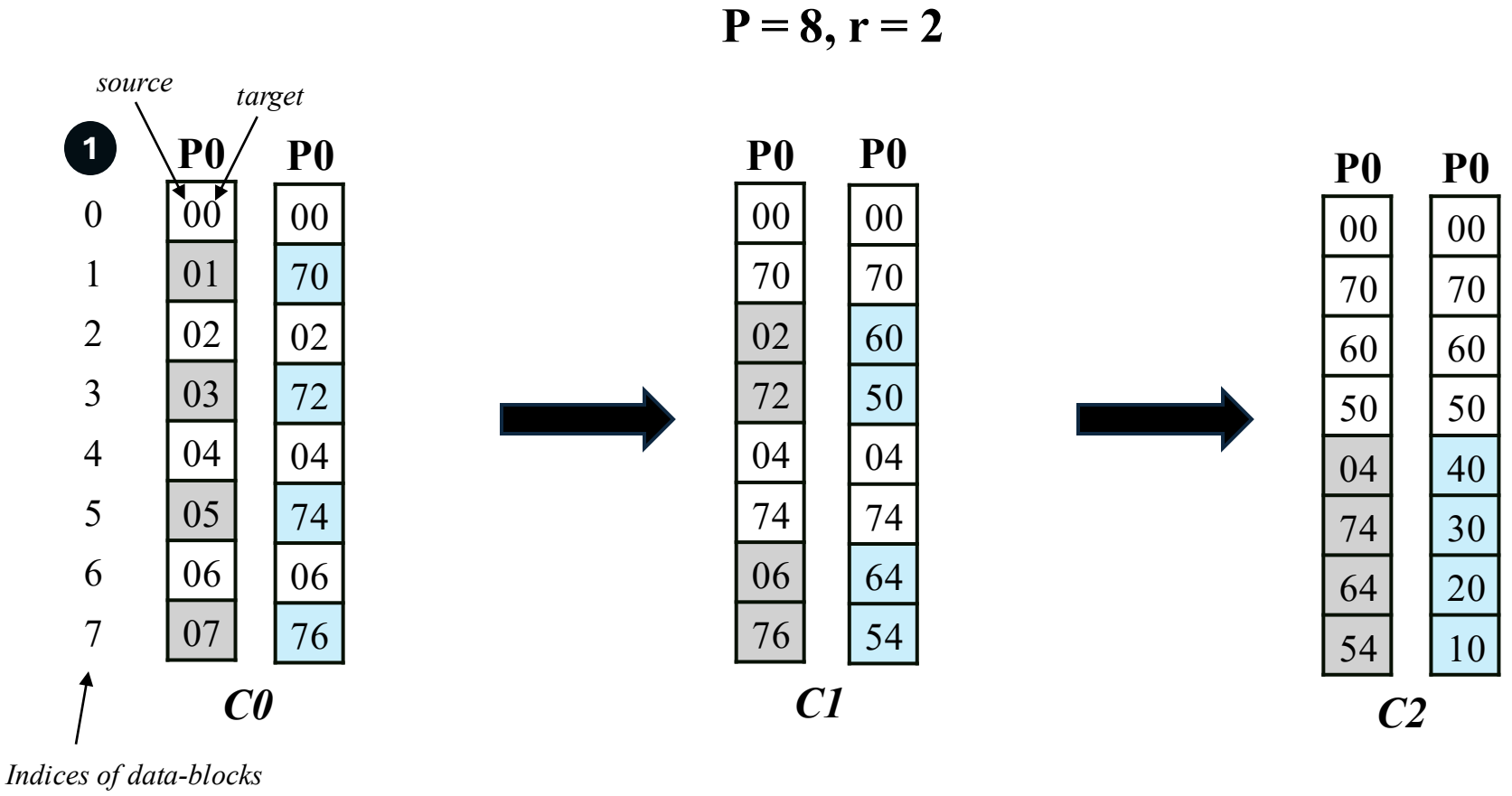
Size of T: 5 x 4 = 20

Size of T ?

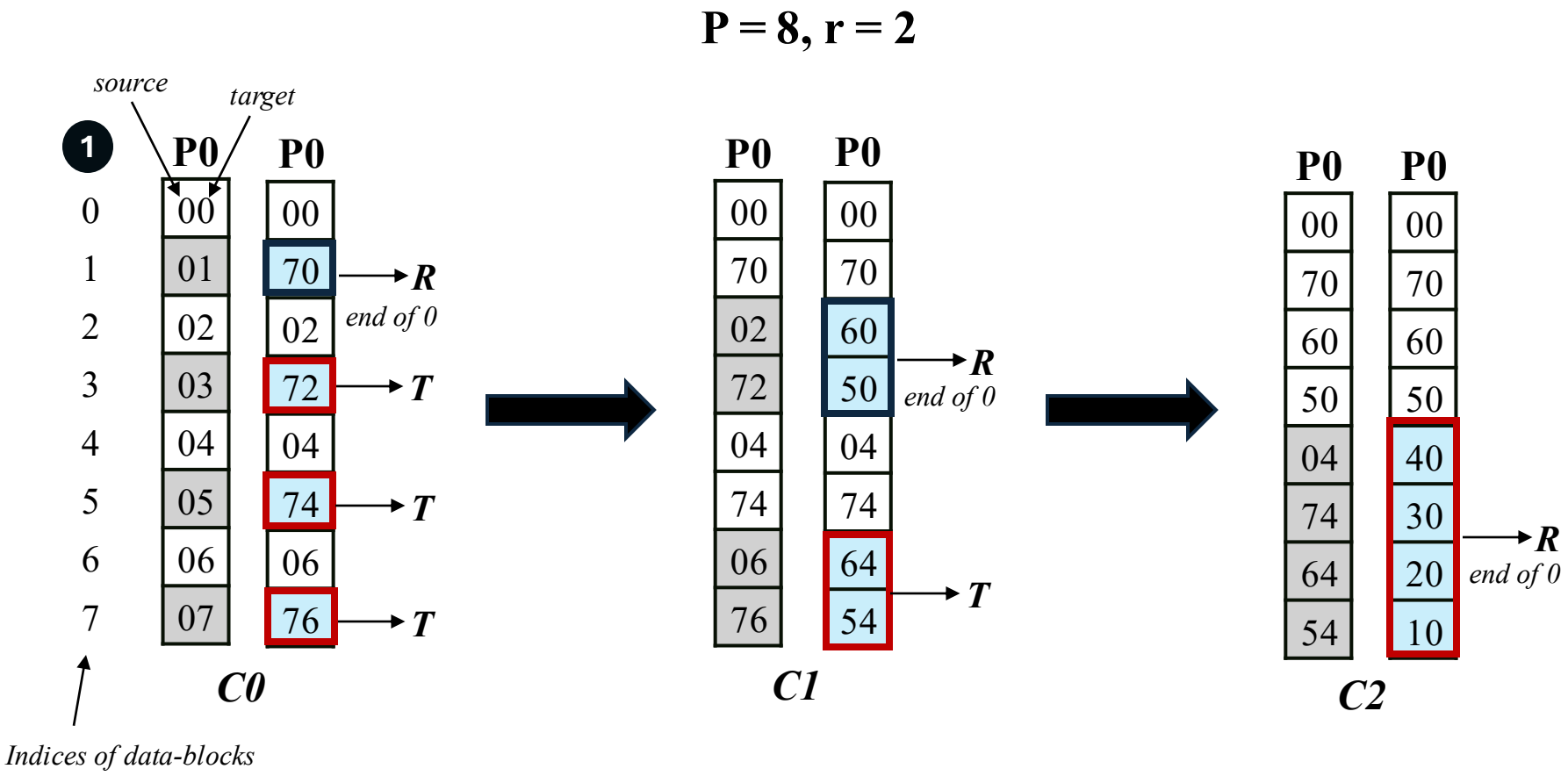


Sparse workloads

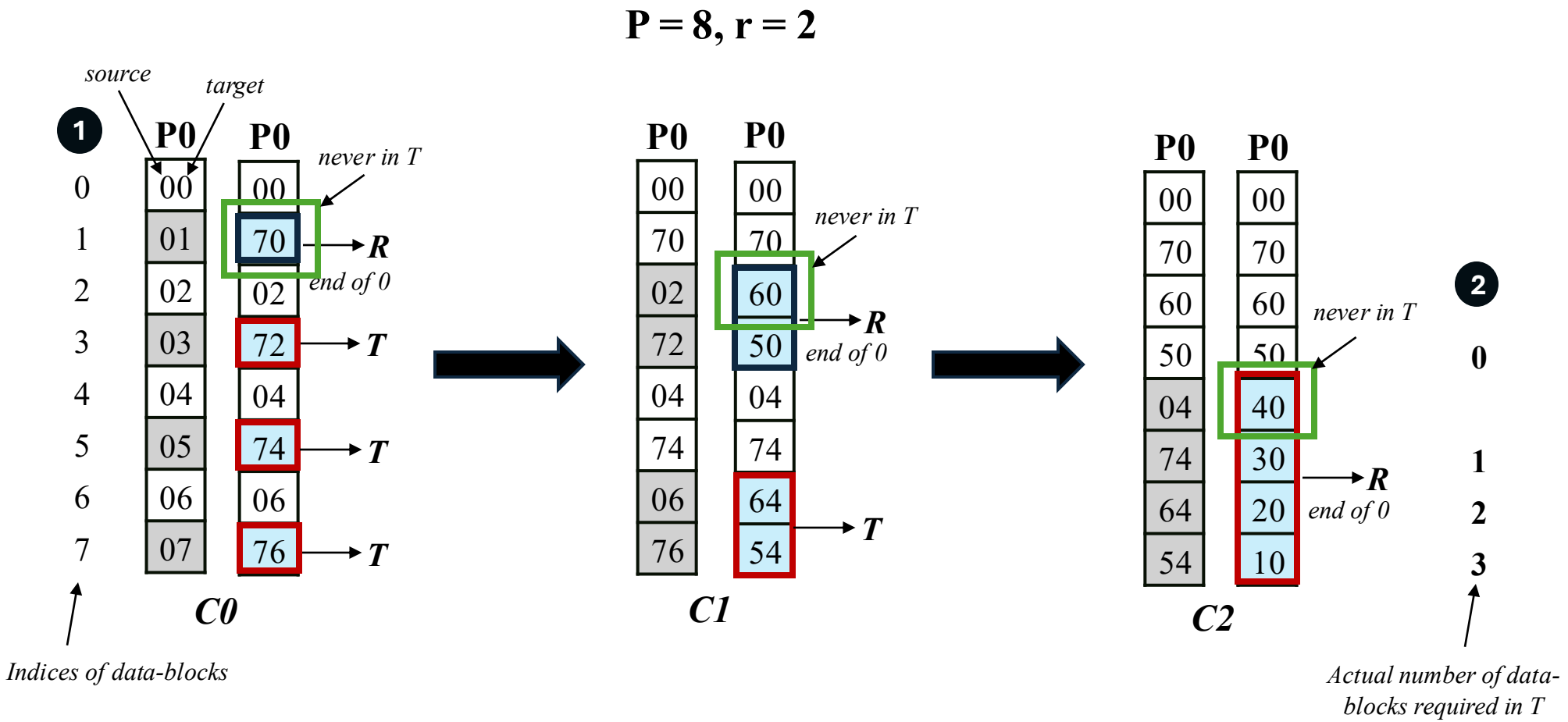
Temporary Buffer Size Analysis



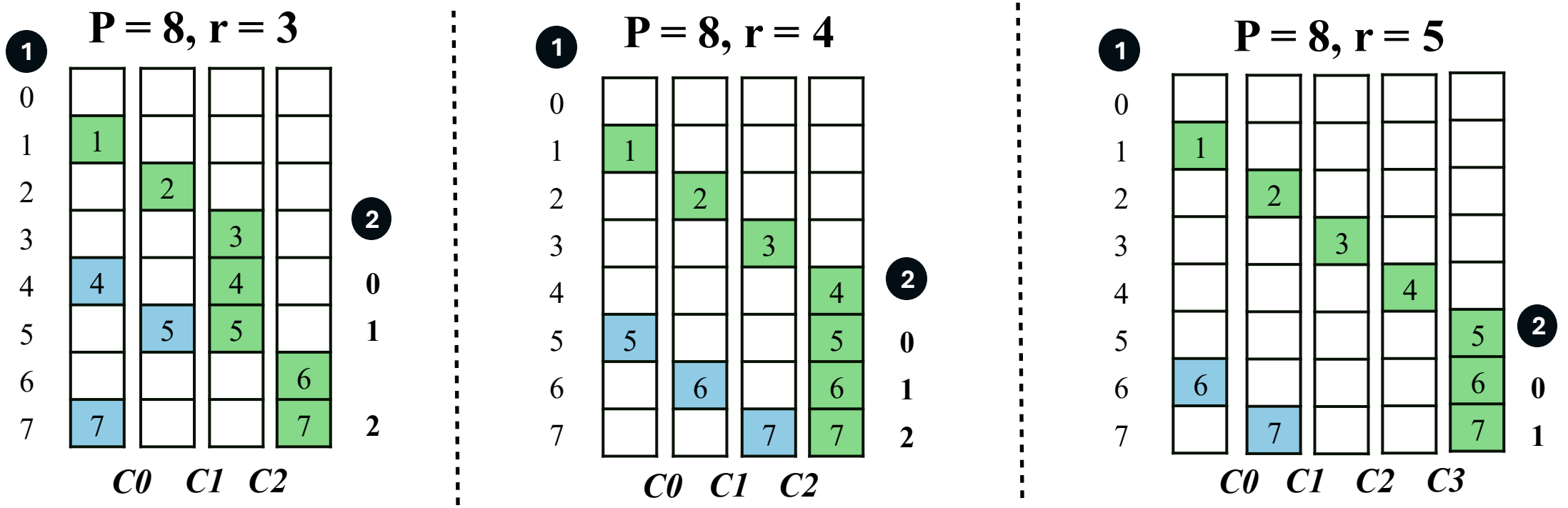
Temporary Buffer Size Analysis



Temporary Buffer Size Analysis



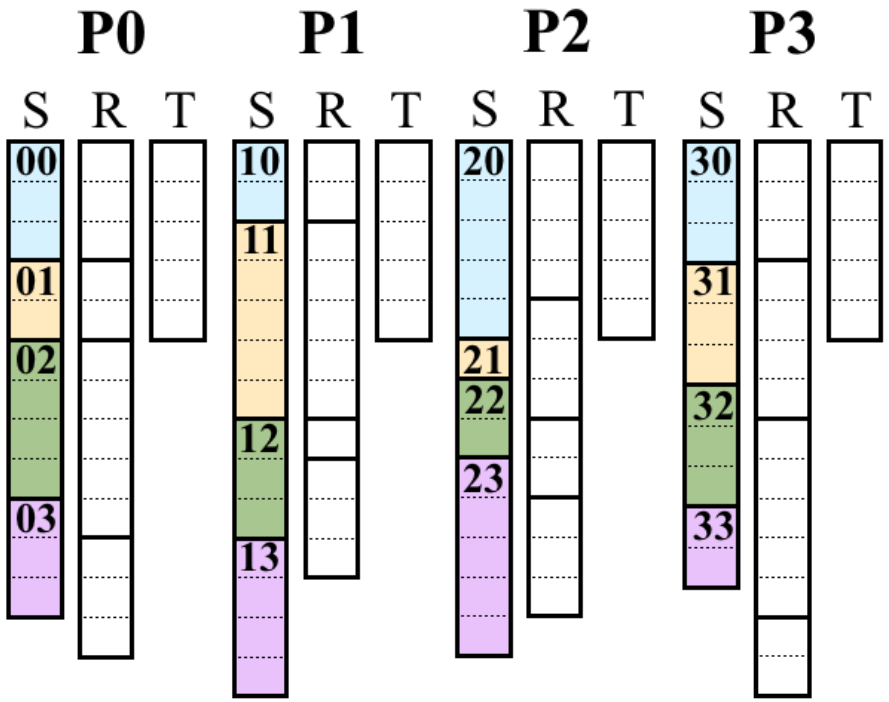
Temporary Buffer Size Analysis



Agenda

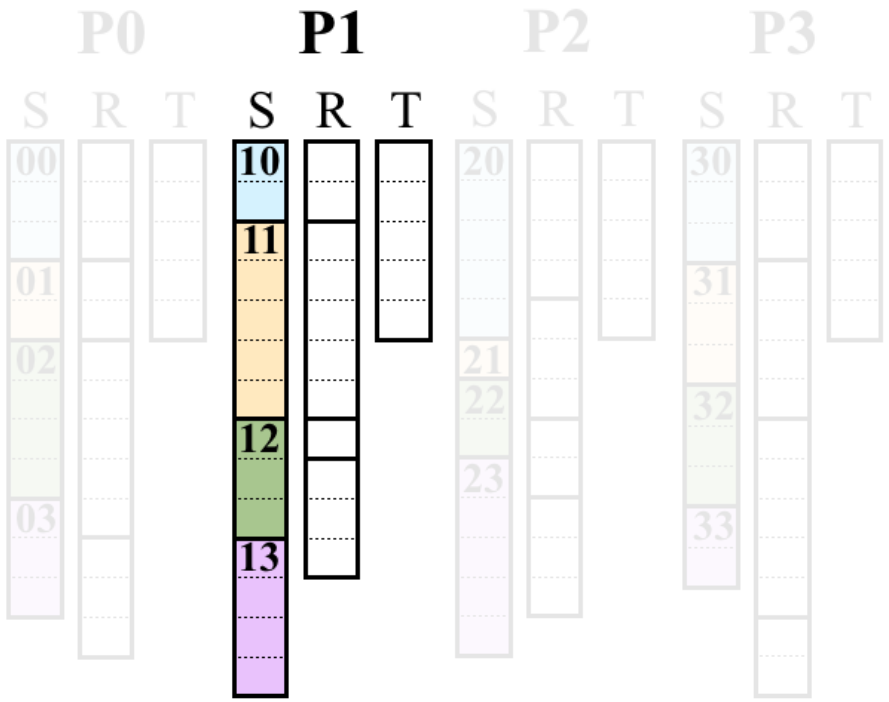
- Introduction and Background
  - Inter-process Communication
  - All-to-all Data Exchange
  - Standard MPI All-to-all Implementations
- Challenges and Solutions
  - Tunability of Performance
  - Logarithmic Non-uniform All-to-all
- Parameterized Logarithmic Algorithm
- Parameterized Linear Algorithm
- Evaluation
- Application
- Conclusion

Example of the ParLogNa with $P = 4$ and $r = 2$.



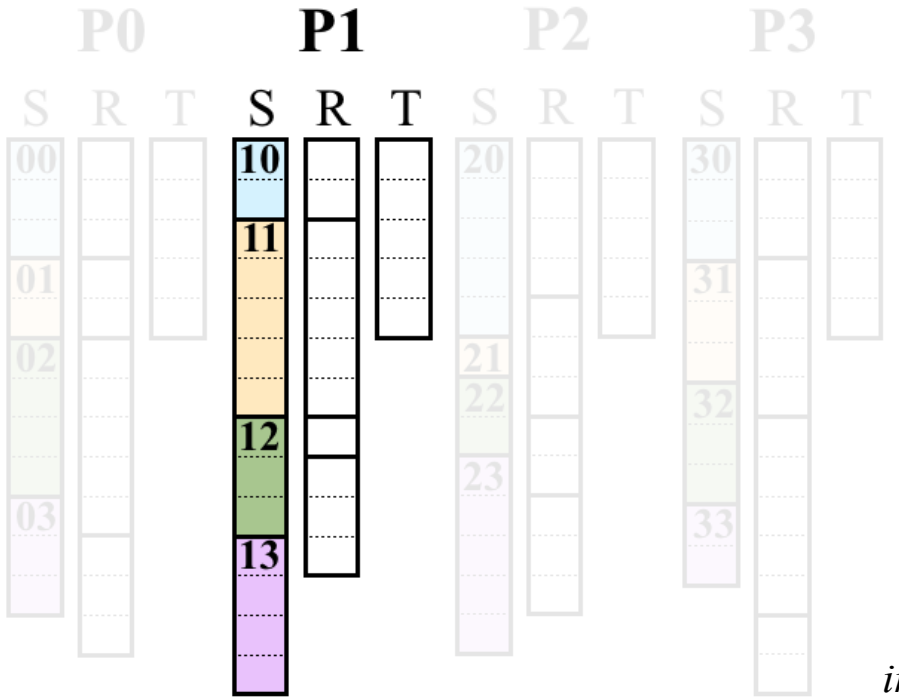
Initial data for 4 process

Example of the ParLogNa with $P = 4$ and $r = 2$.

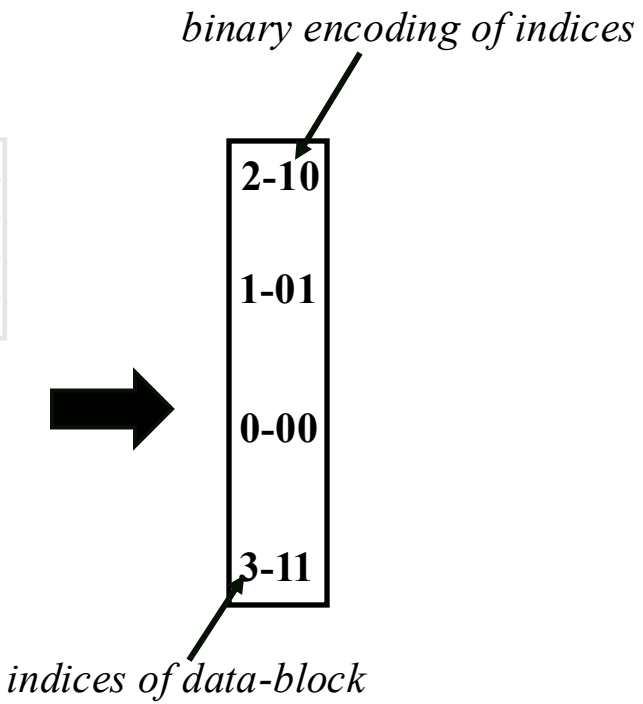


Taking P1 for example

Example of the ParLogNa: Rotation Array and Binary Encoding

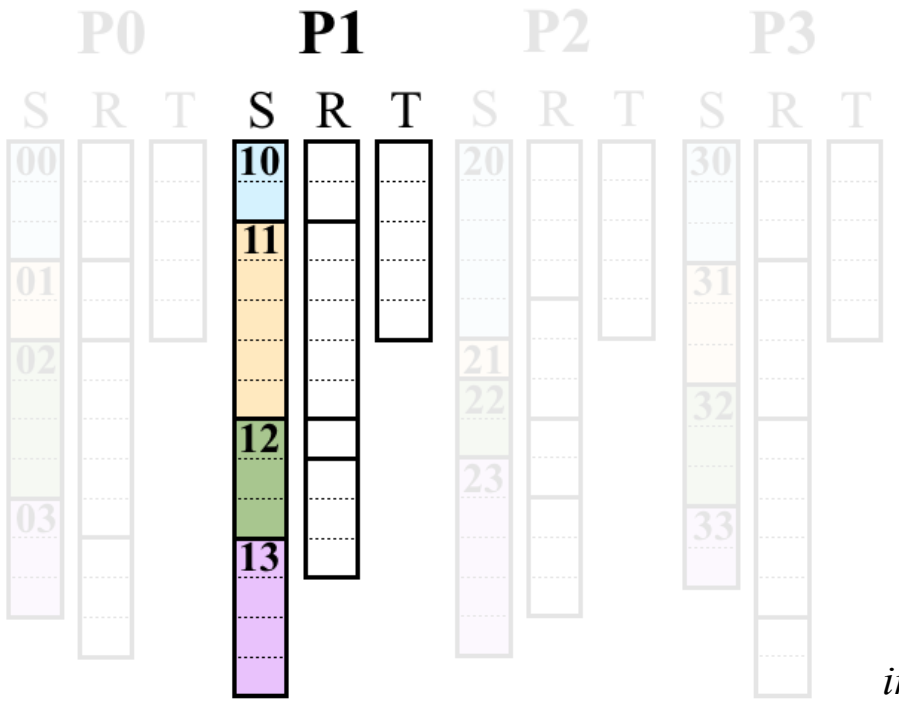


Taking P1 for example

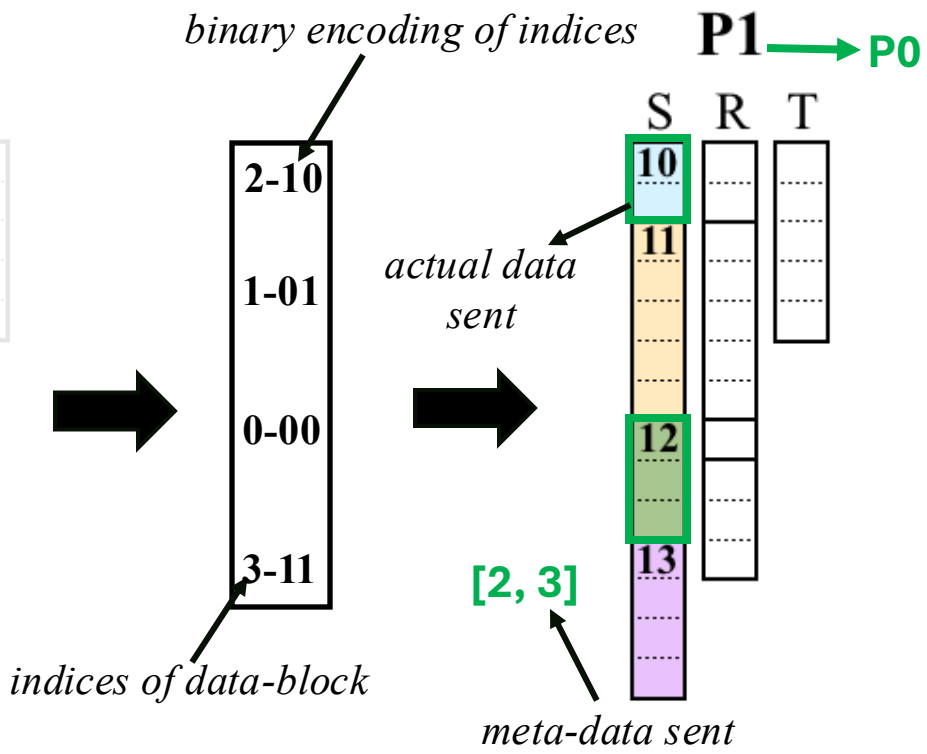


Rotation array

Example of the ParLogNa: Comm Round 0

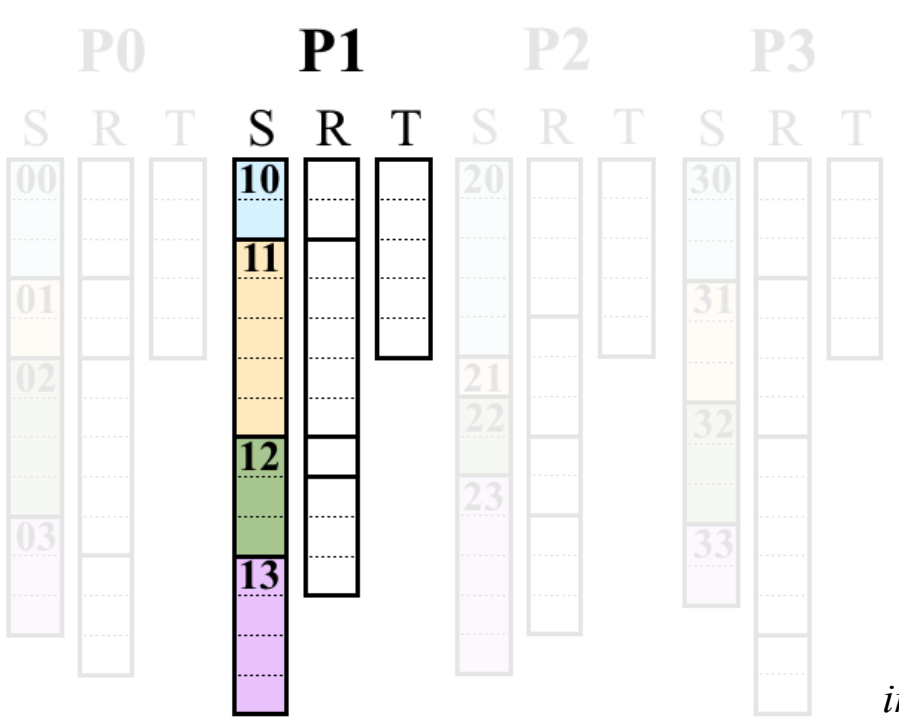


Taking P1 for example

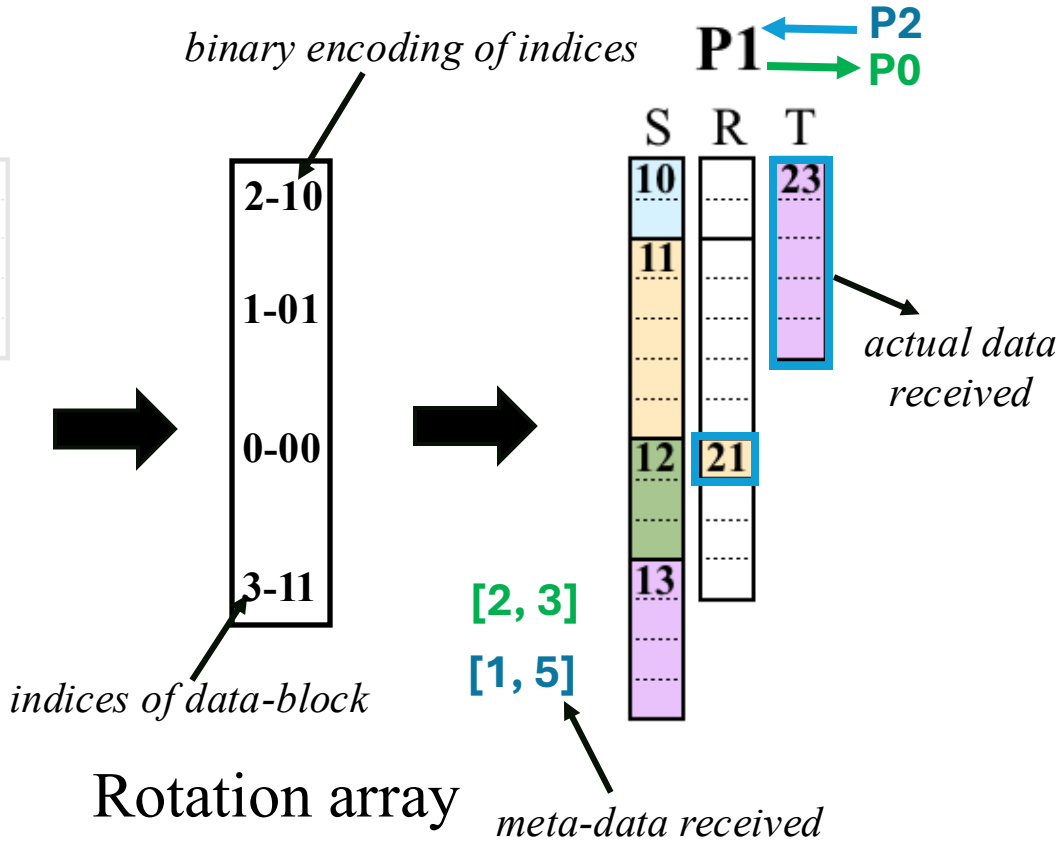


Rotation array

Example of the ParLogNa: Comm Round 0

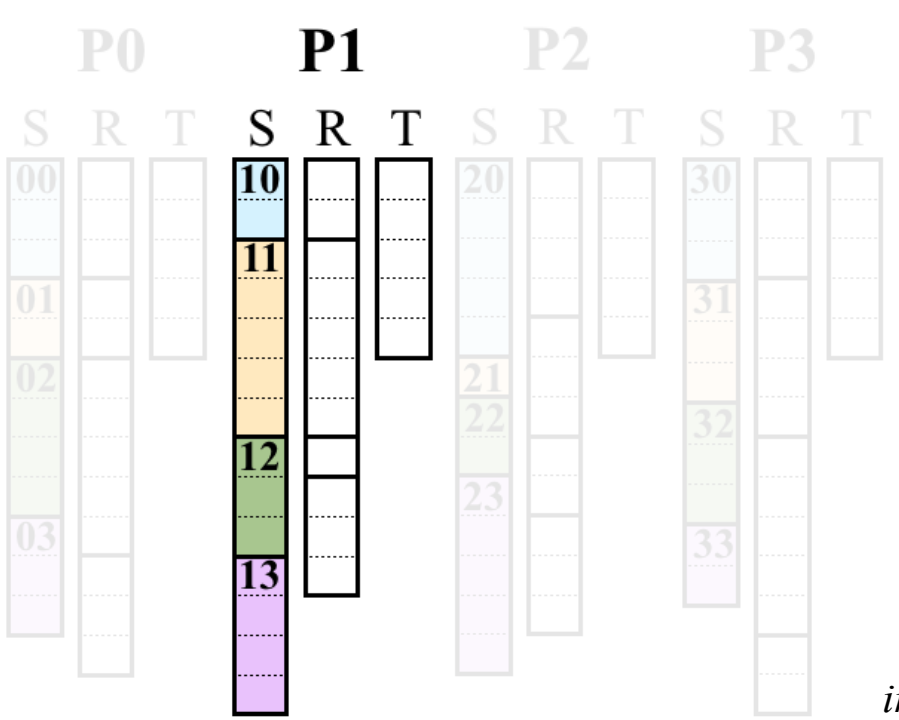


Taking P1 for example

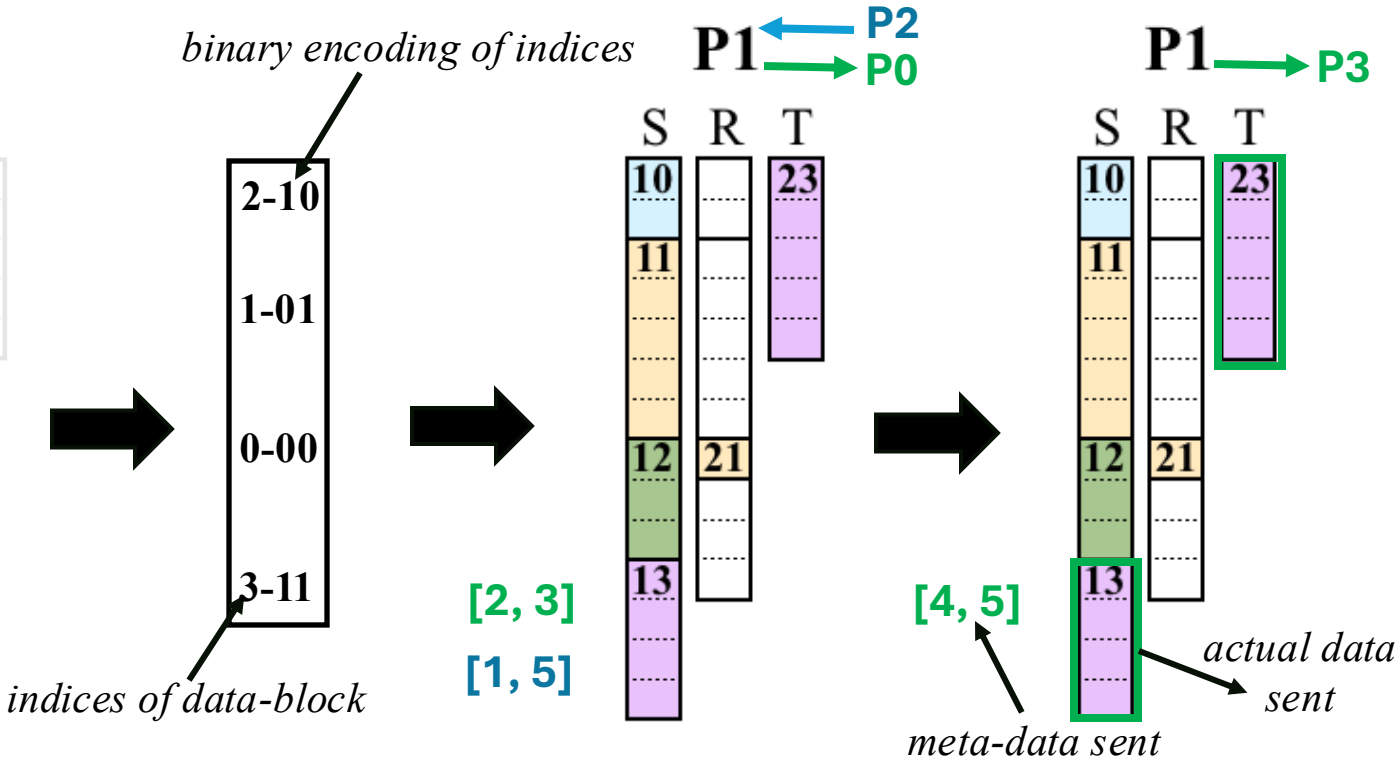


Rotation array

Example of the ParLogNa: Comm Round 1

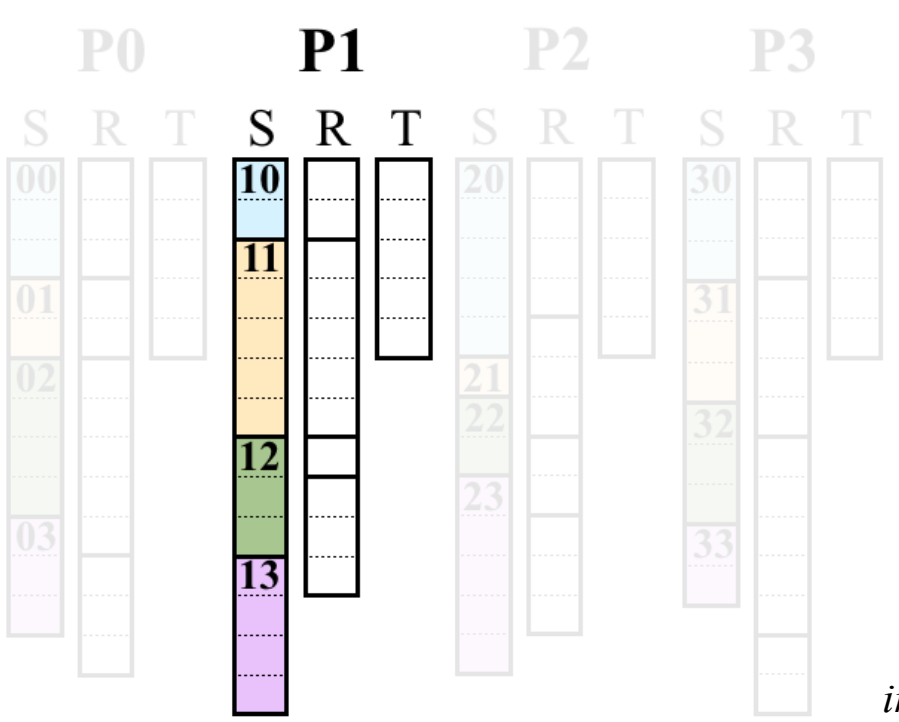


Taking P1 for example

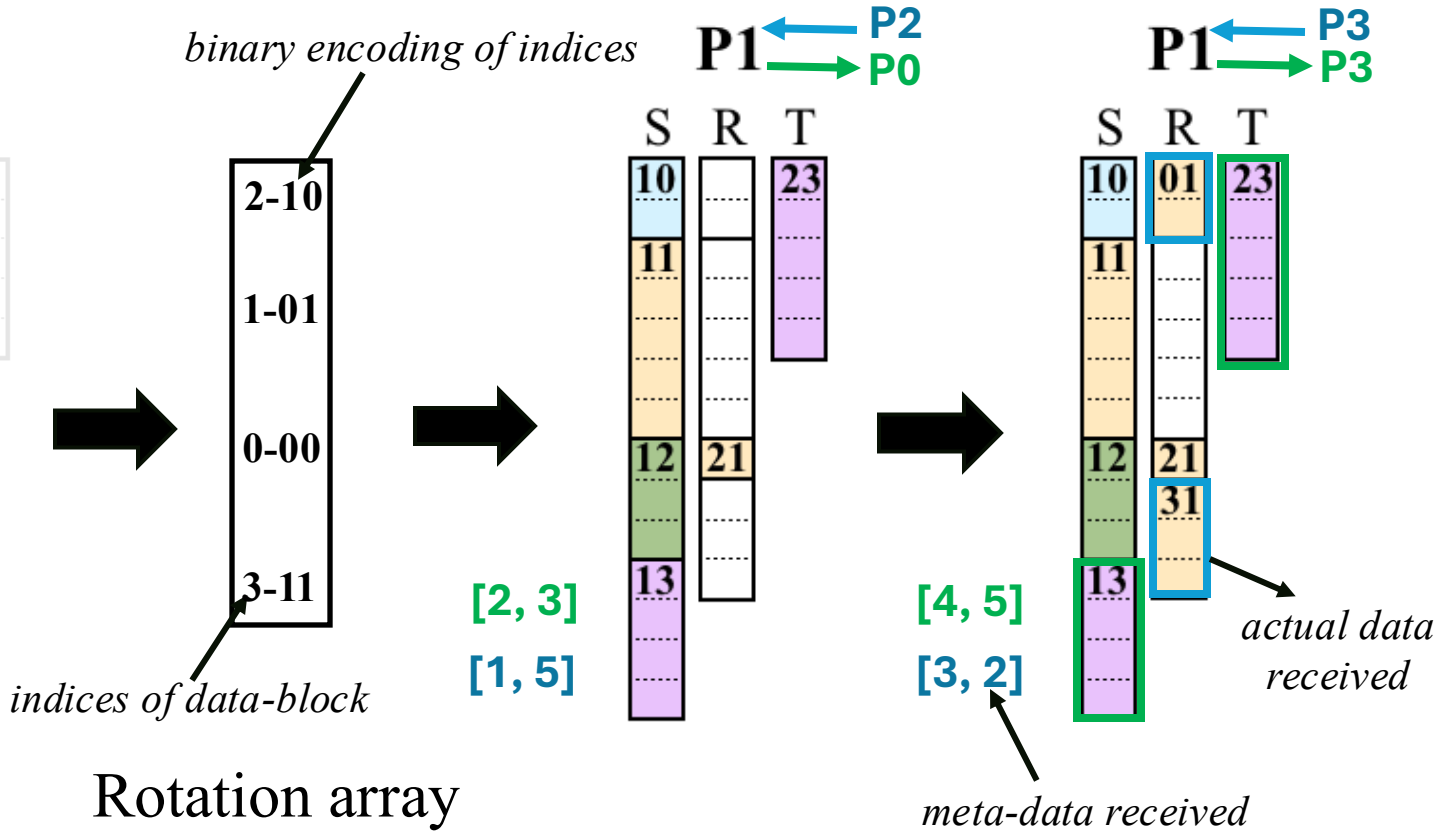


Rotation array

Example of the ParLogNa: Comm Round 1



Taking P1 for example

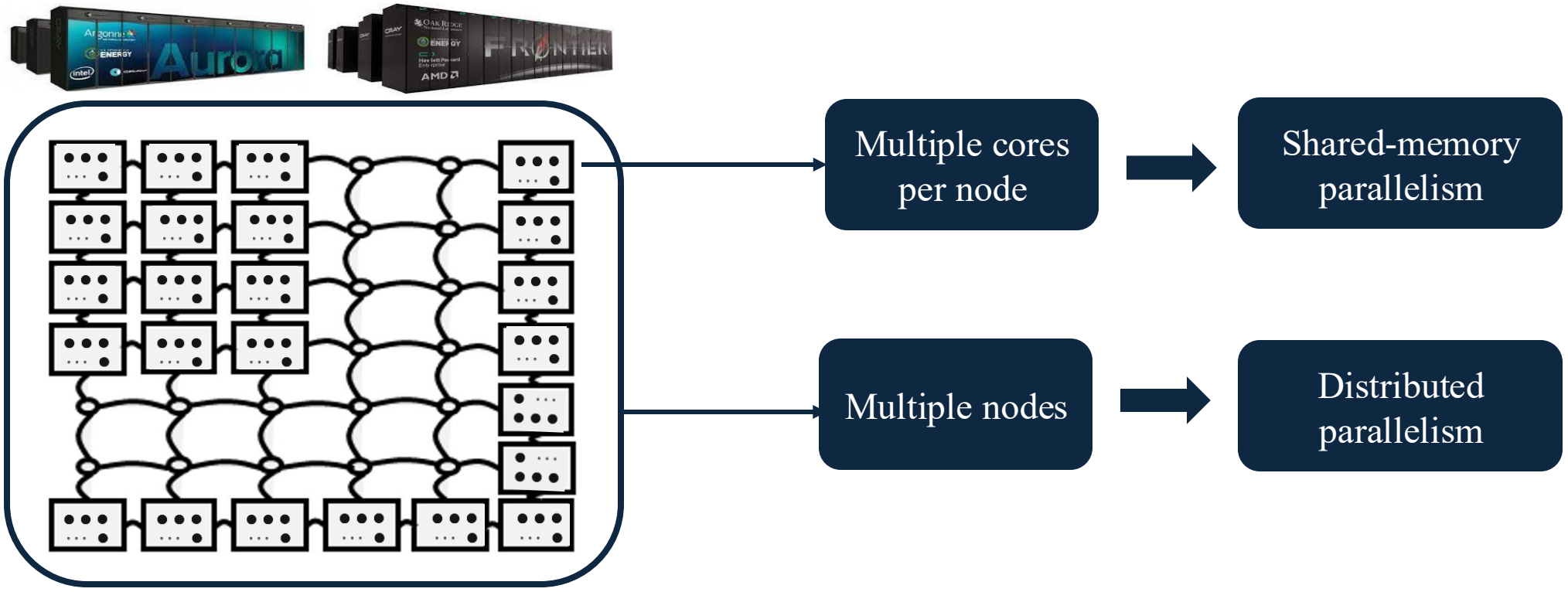


Rotation array

Agenda

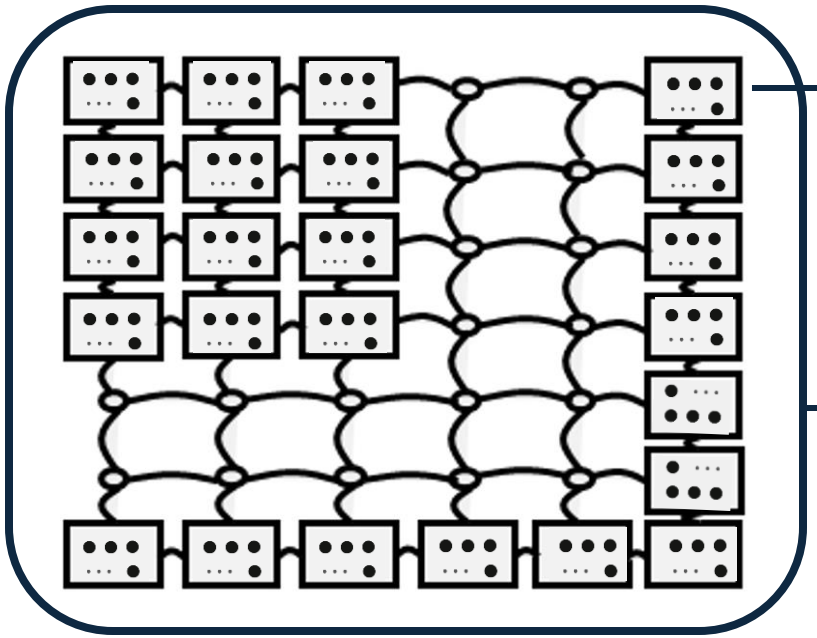
- Introduction and Background
  - Inter-process Communication
  - All-to-all Data Exchange
  - Standard MPI All-to-all Implementations
- Challenges and Solutions
  - Tunability of Performance
  - Logarithmic Non-uniform All-to-all
- Parameterized Logarithmic Algorithm
- Parameterized Linear Algorithm
- Evaluation
- Application
- Conclusion

Architecture of Modern HPC Systems



Modern HPC systems

Architecture of Modern HPC Systems

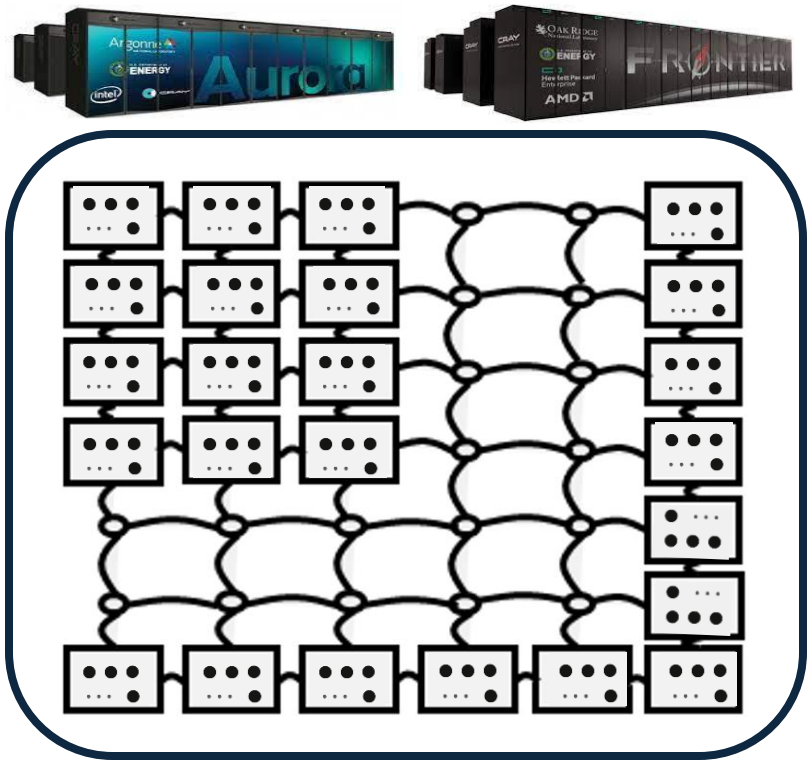


Modern HPC systems

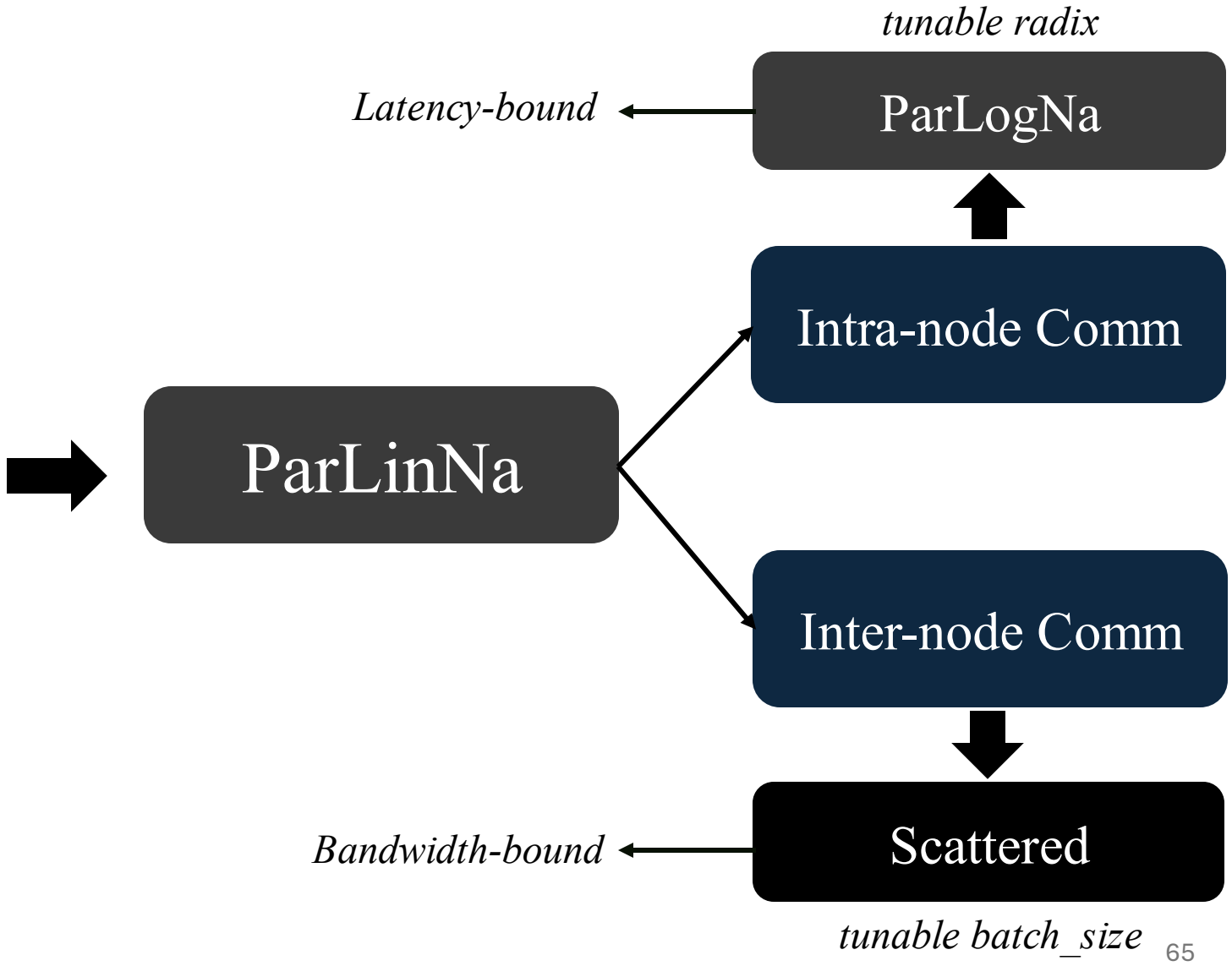
Multiple cores per node → Shared-memory parallelism → faster

Multiple nodes → Distributed parallelism

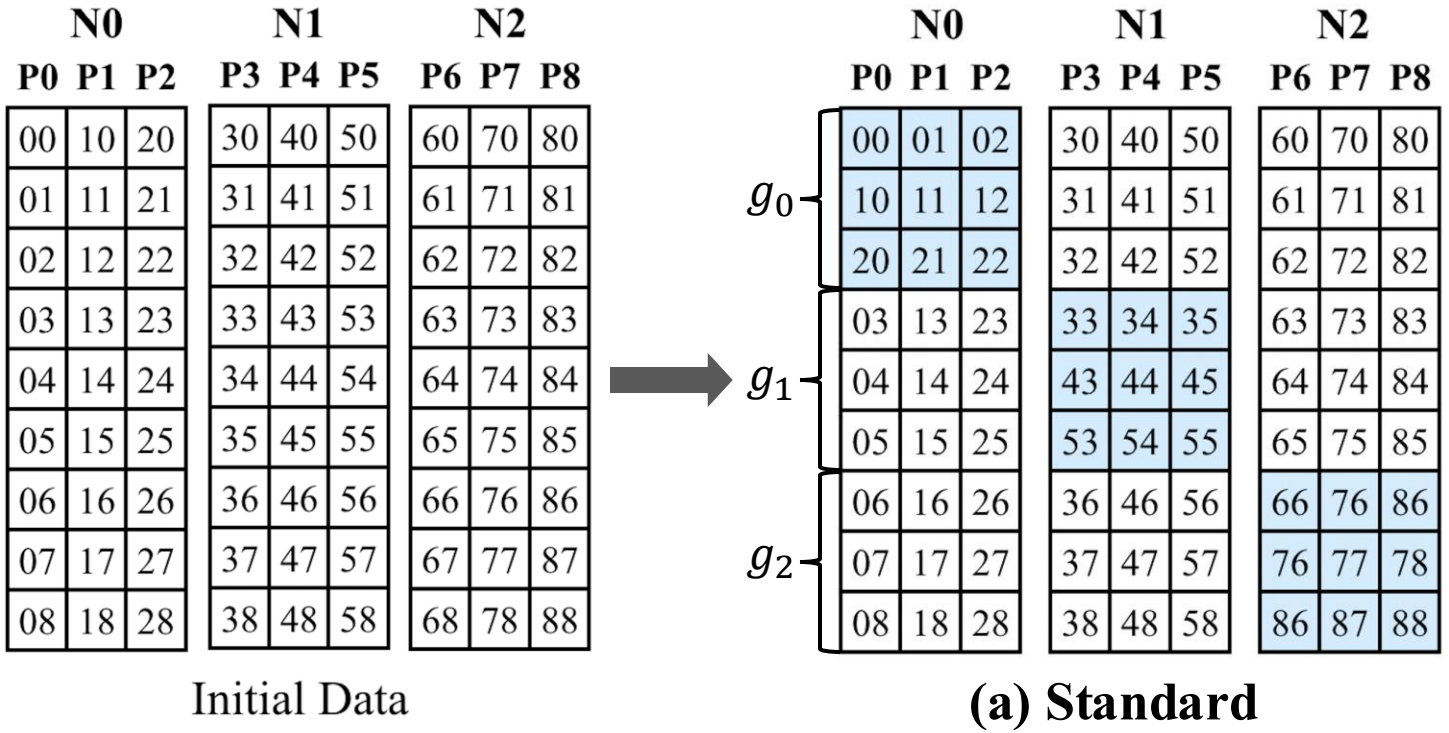
Parameterized Linear Non-uniform All-to-all (ParLinNa)



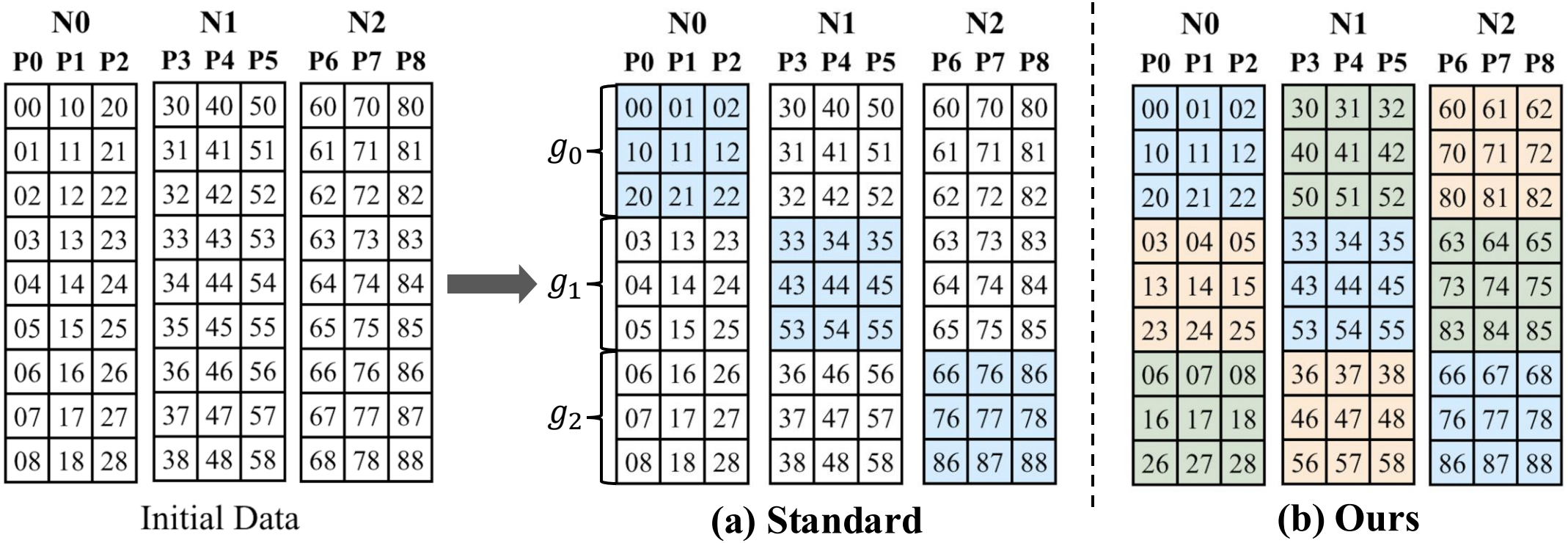
Modern HPC systems



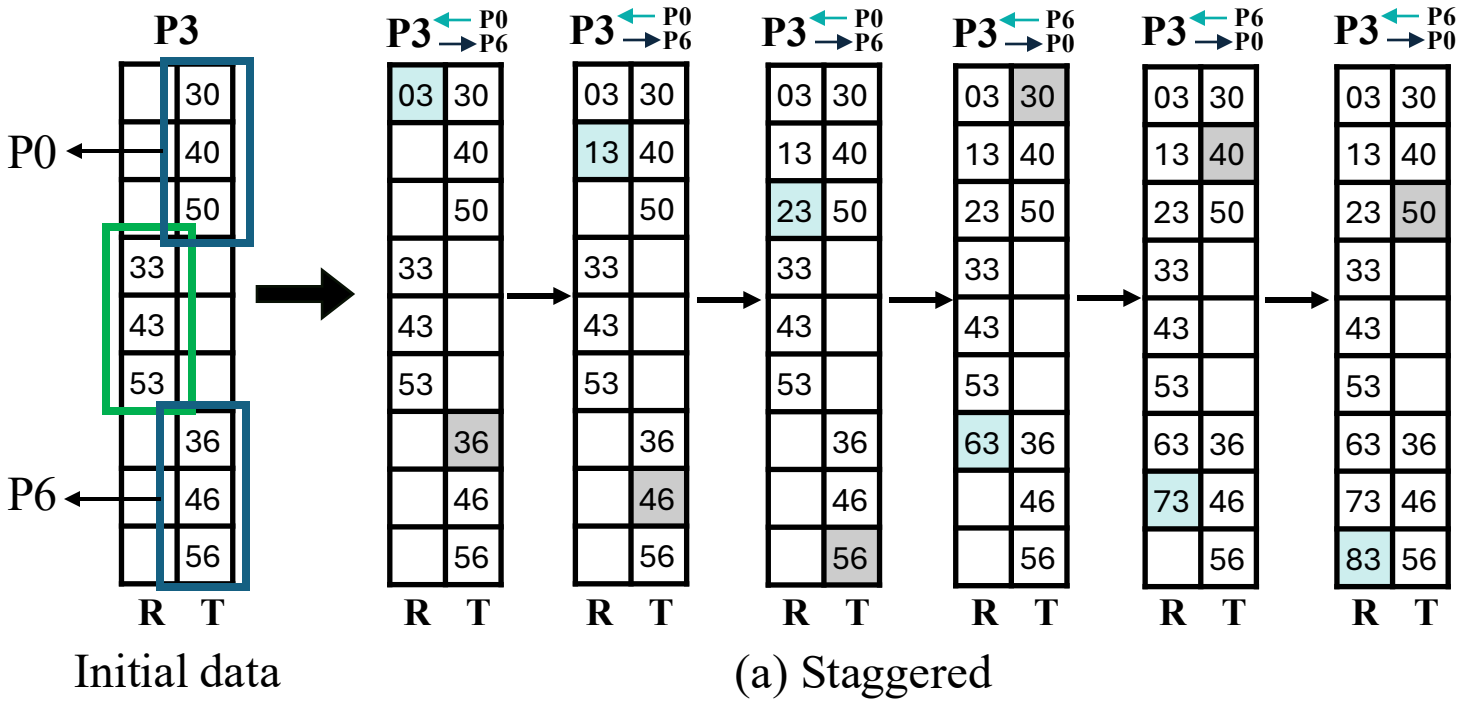
Intra-node Communication: Two Strategies



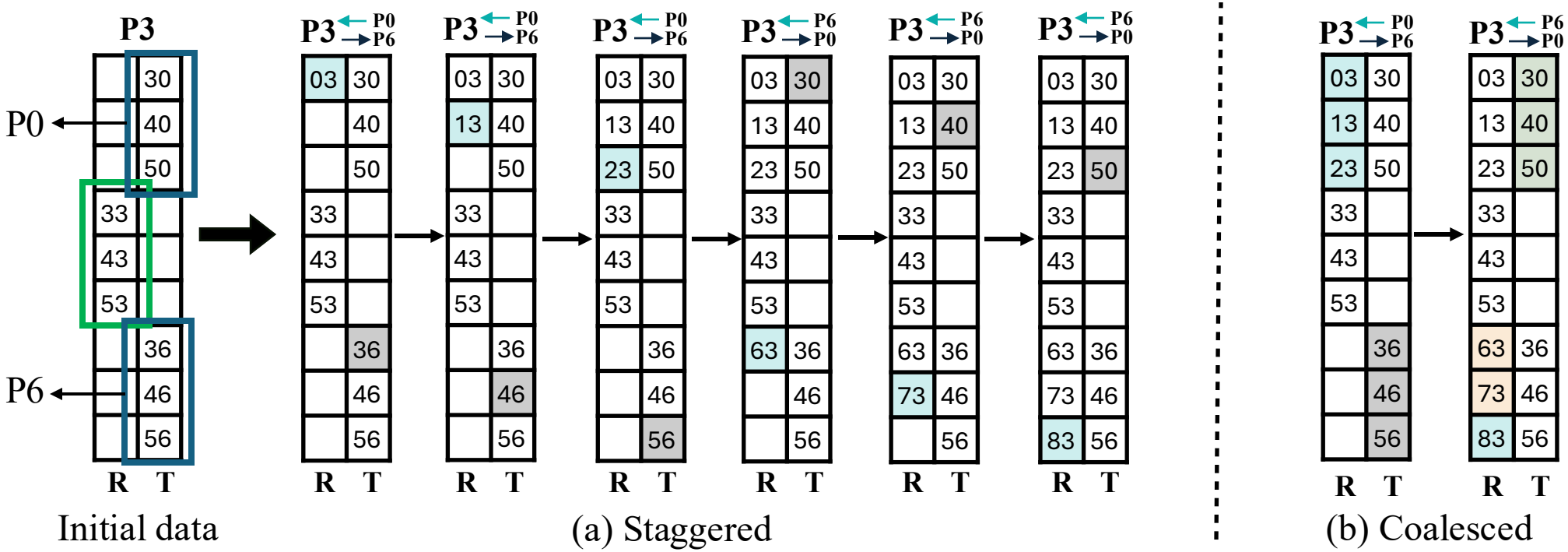
Intra-node Communication: Two Strategies



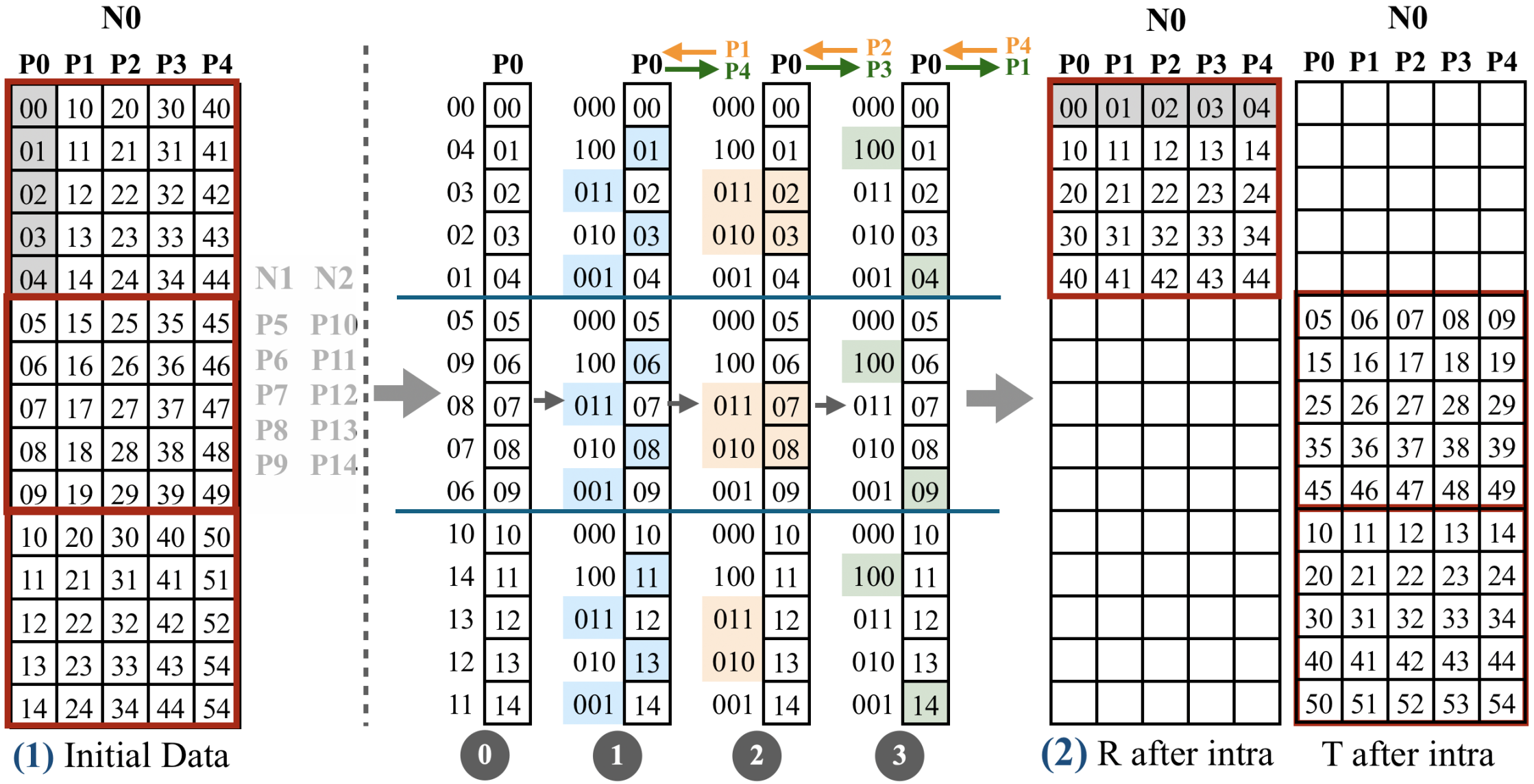
Inter-node Communication: Two Patterns



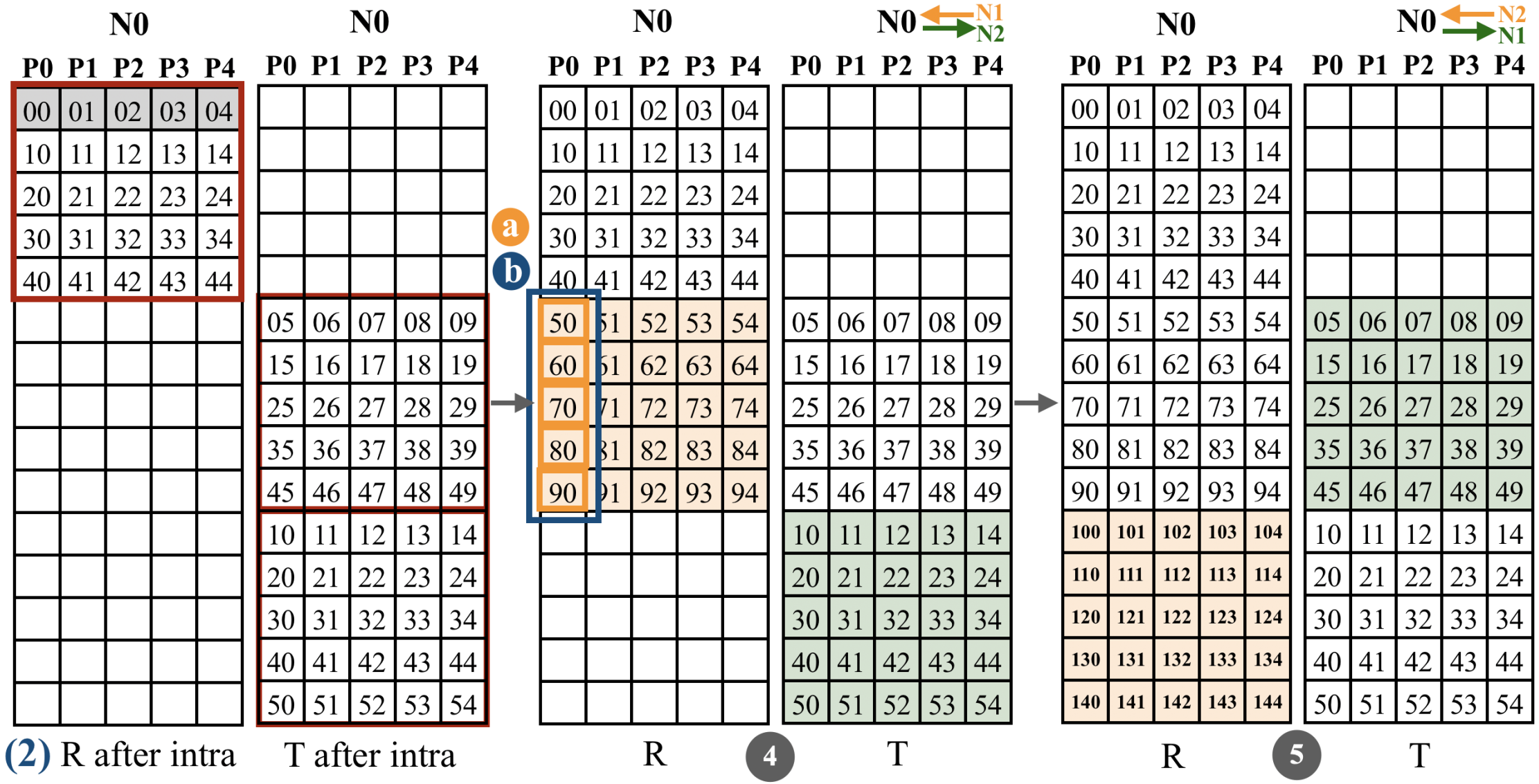
Inter-node Communication: Two Patterns



Intra-node Communication Example



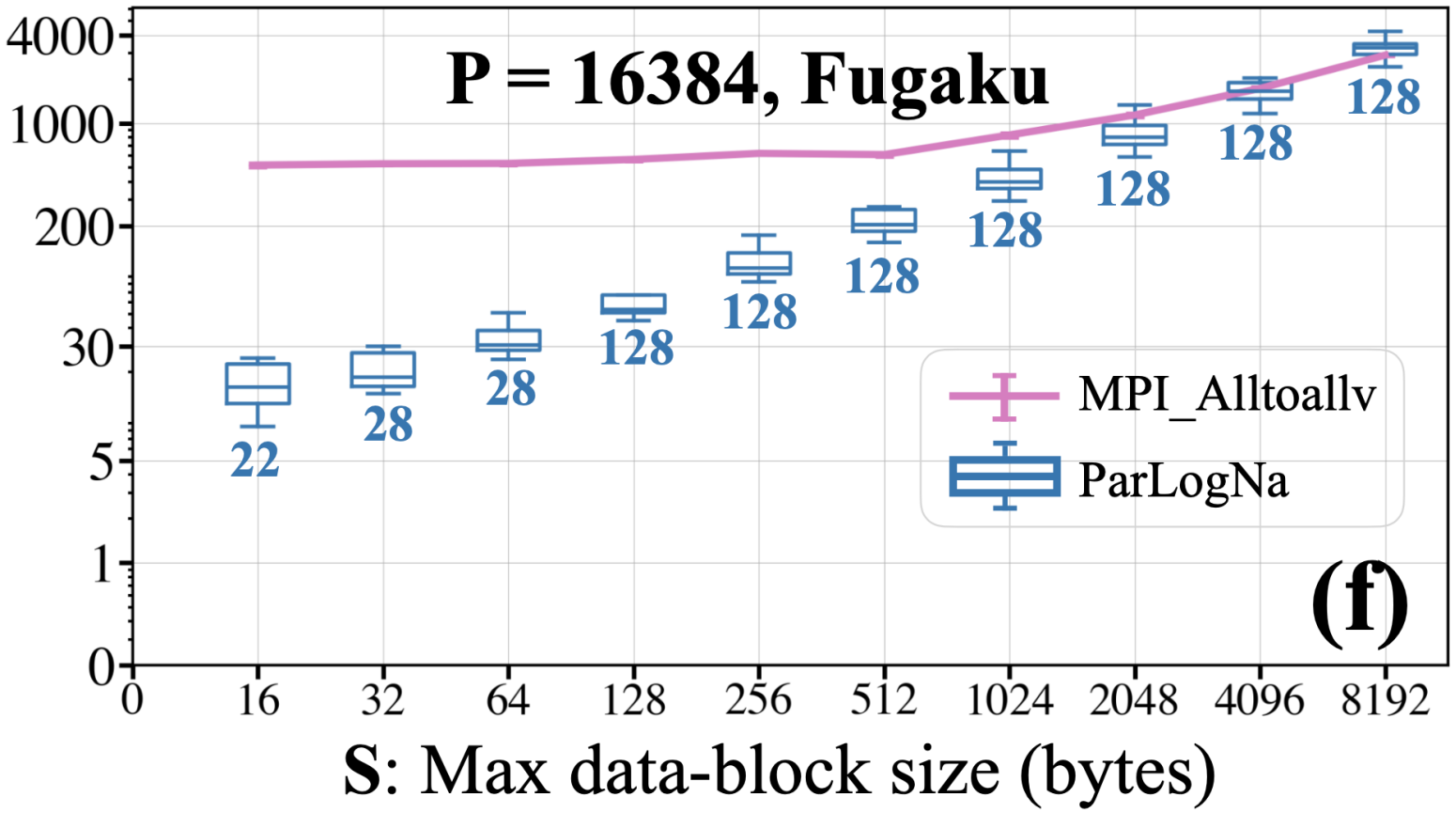
Inter-node Communication Example



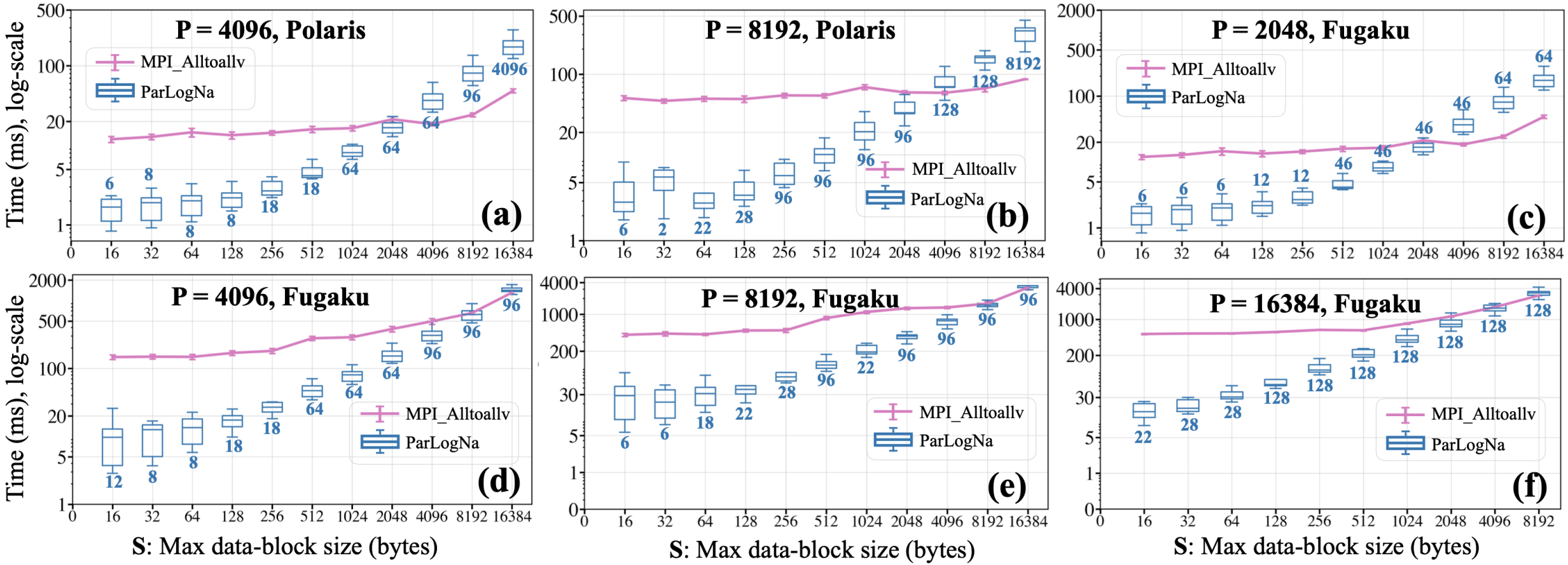
Agenda

- Introduction and Background
  - Inter-process Communication
  - All-to-all Data Exchange
  - Standard MPI All-to-all Implementations
- Challenges and Solutions
  - Tunability of Performance
  - Logarithmic Non-uniform All-to-all
- Parameterized Logarithmic Algorithm
- Parameterized Linear Algorithm
- Evaluation
- Application
- Conclusion

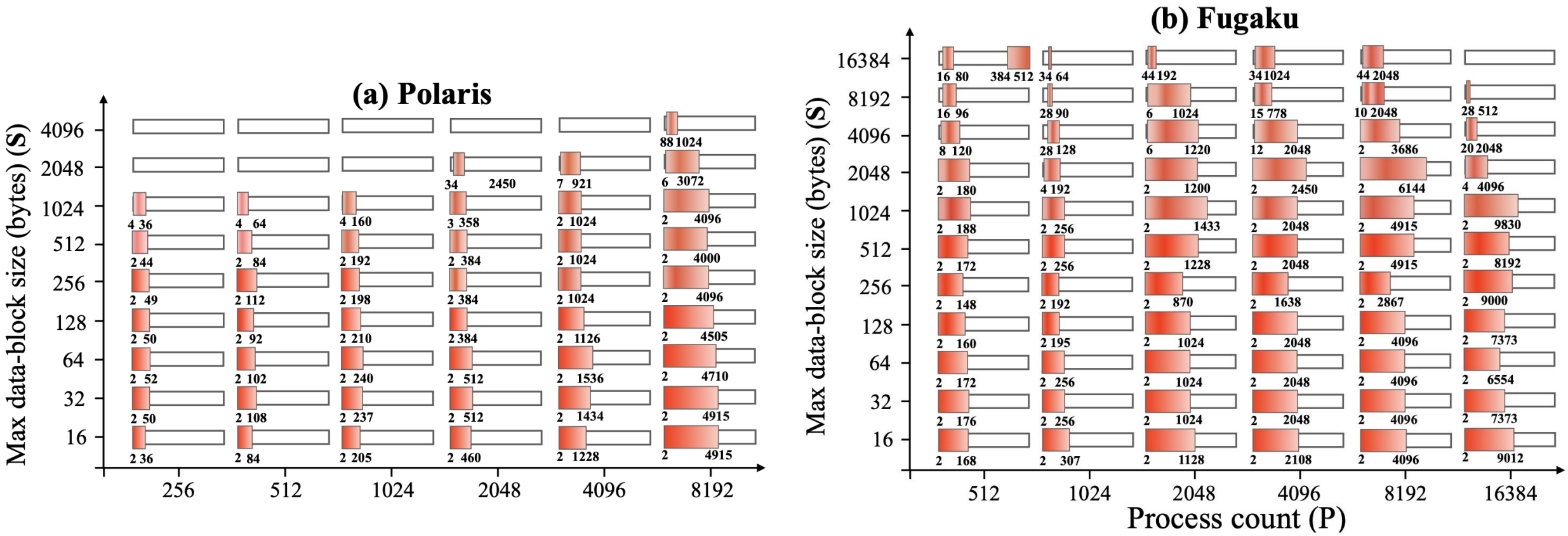
Comparing ParLogNa with MPI_Alltoallv



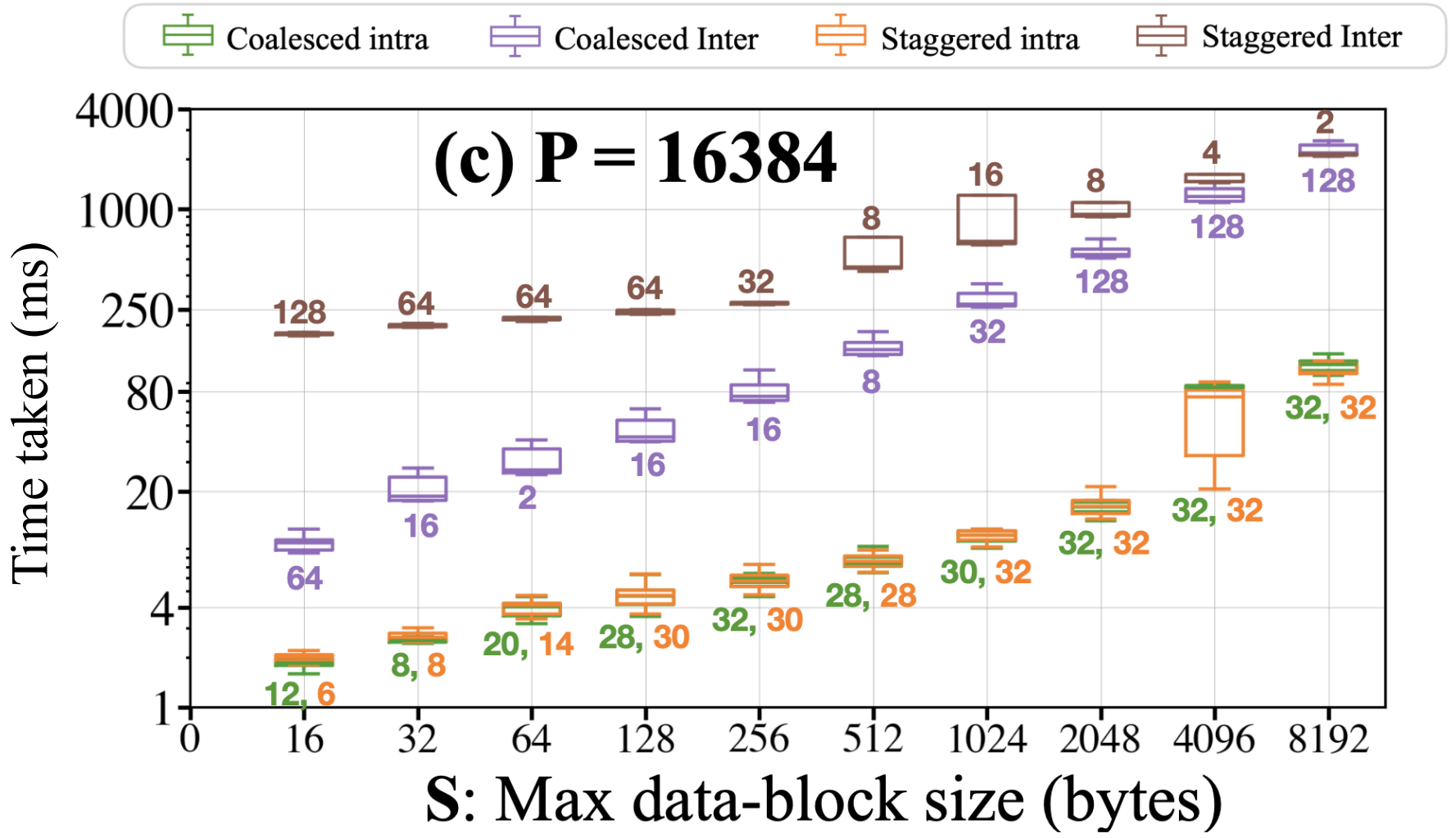
Comparing ParLogNa with MPI_Alltoallv



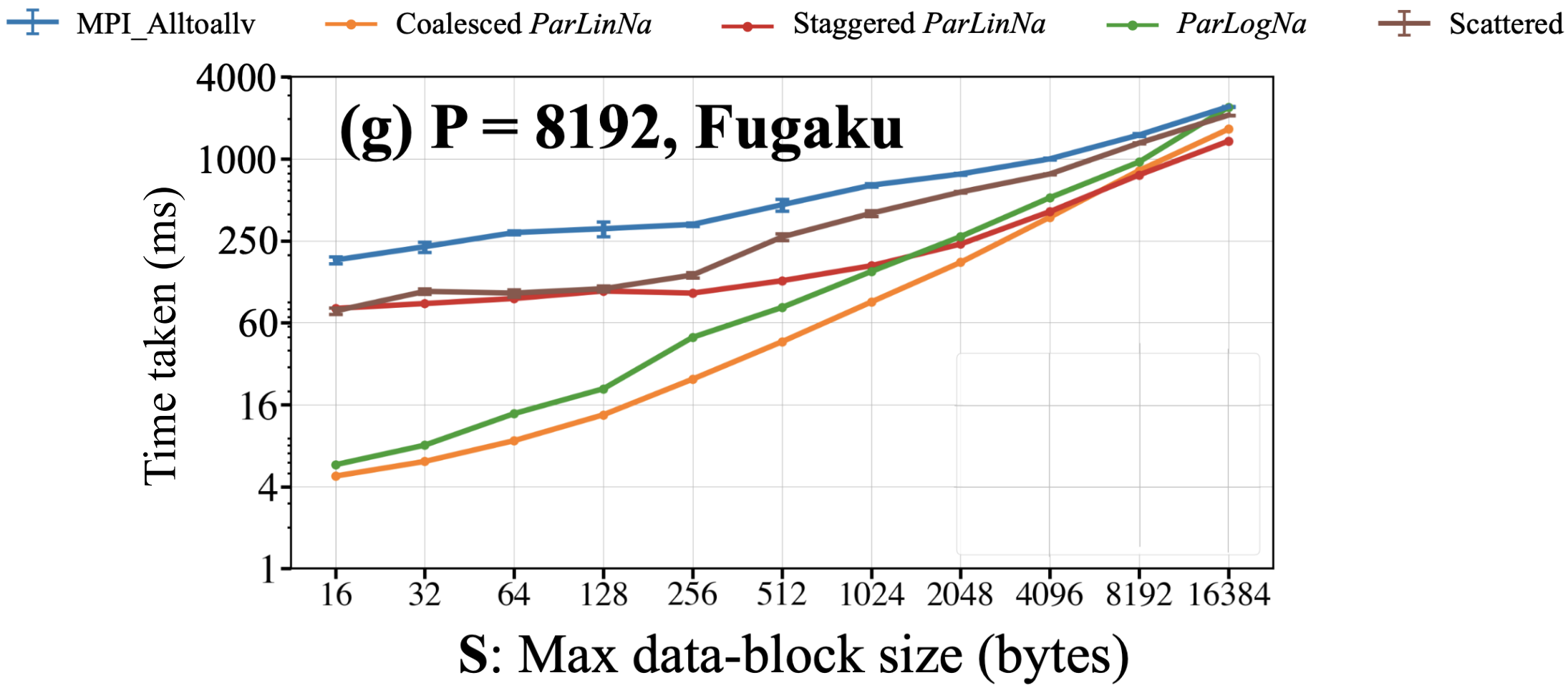
Ranges of radix where ParLogNa outperforms MPI_Alltoallv



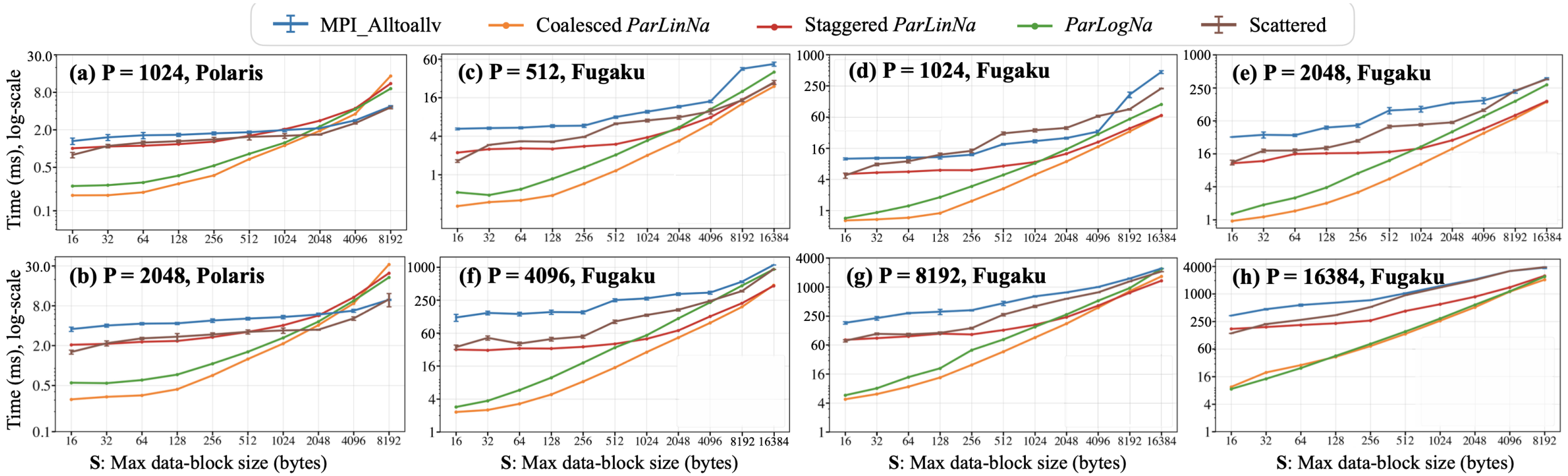
Comparing Coalesced and Staggered ParLinNa



Comparing Proposed Algorithms with the Top-performing Benchmark



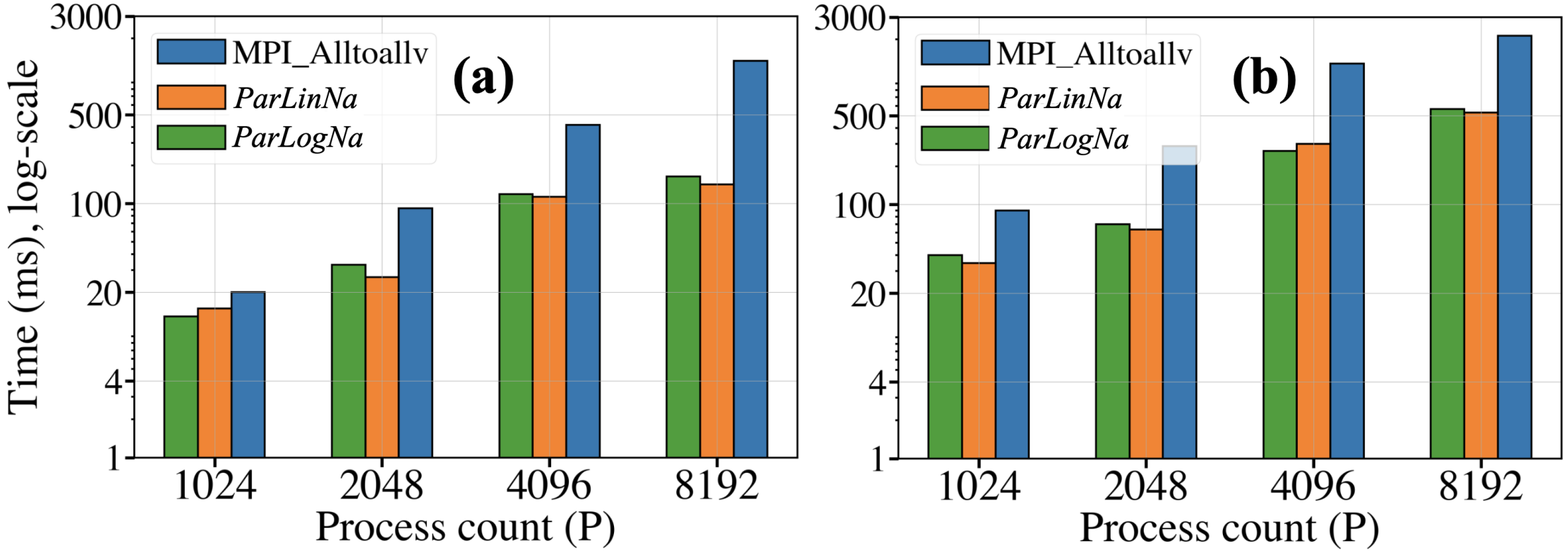
Comparing Proposed Algorithms with the Top-performing Benchmark



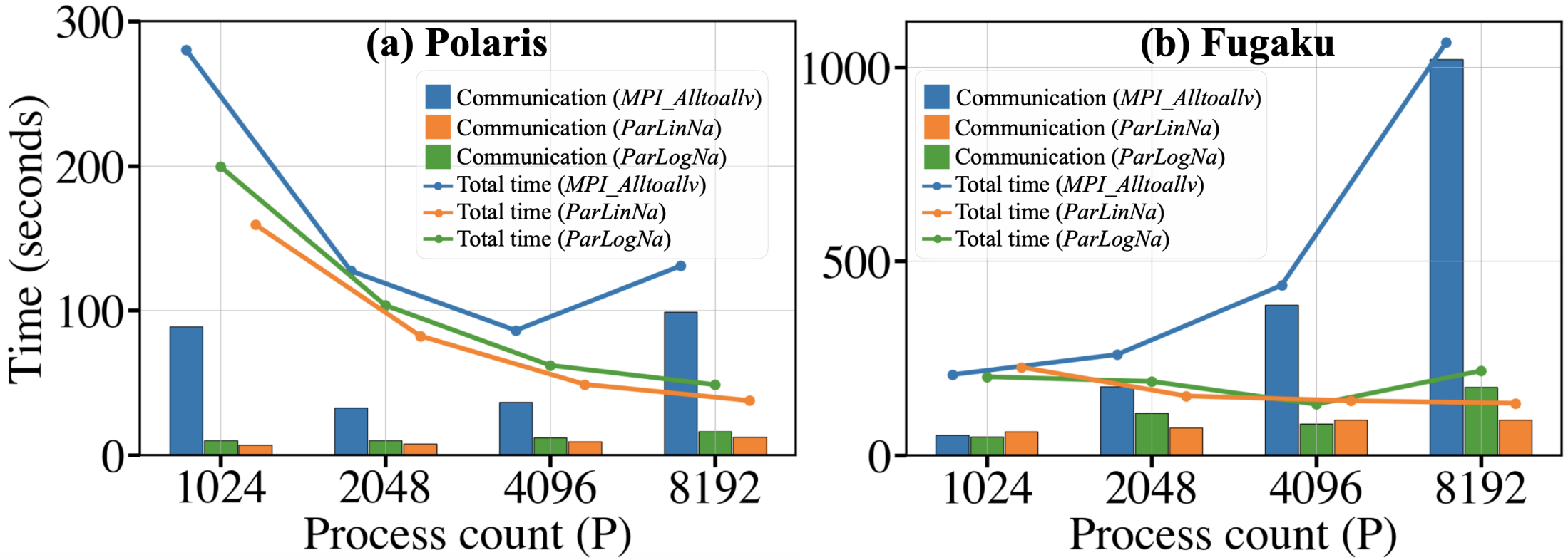
Agenda

- Introduction and Background
  - Inter-process Communication
  - All-to-all Data Exchange
  - Standard MPI All-to-all Implementations
- Challenges and Solutions
  - Tunability of Performance
  - Logarithmic Non-uniform All-to-all
- Parameterized Logarithmic Algorithm
- Parameterized Linear Algorithm
- Evaluation
- Application
- Conclusion

Performance of Applying Our Algorithms to FFT



Performance of Applying Our Algorithms to Path Finding



Agenda

- Introduction and Background
  - Inter-process Communication
  - All-to-all Data Exchange
  - Standard MPI All-to-all Implementations
- Challenges and Solutions
  - Tunability of Performance
  - Logarithmic Non-uniform All-to-all
- Parameterized Logarithmic Algorithm
- Parameterized Linear Algorithm
- Evaluation
- Application
- Conclusion

Conclusion

- We have presented algorithms (ParLogNa and ParLinNa) that can improve an important class of collective functions.
- The work can help vendors further optimize their collective routines.
- The work can have a direct impact on a range of applications that uses all to all communication.

Thank you!