Enhancing Intra-Node GPU-to-GPU Performance in MPI+UCX through Multi-Path Communication

AmirHossein Sojoodi, Yiltan Hassan Temucin, and Ahmad Afsahi

Parallel Processing Research Laboratory (PPRL),
Department of Electrical and Computer Engineering

Smith Engineering
Queen's University, Kingston, Canada

ExHET 2024,
Scotland, Edinburgh, March 2nd, 2024

Introduction

Background

Design and Evaluation

Conclusion

1. HPC and Cluster Computing
2. Age of Memory Hungry Applications
3. GPUs and High-Bandwidth Interconnects
4. Multi-Path Communication
5. Why? And why not?

Multi-Path Communication

- Utilize available interconnects to transfer data more efficiently

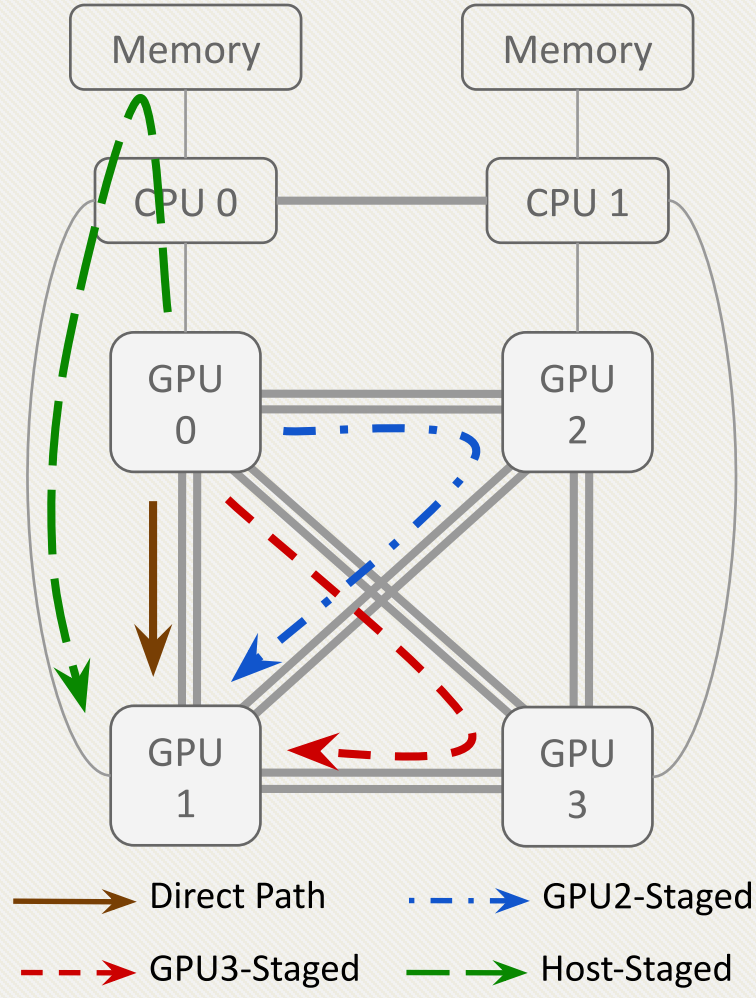


Figure 1: A multi-GPU data transfer

1. MPI and UCX
2. GPUs and GPU-enabled Clusters
3. Bidirectional Data Transfer

MPI and UCX

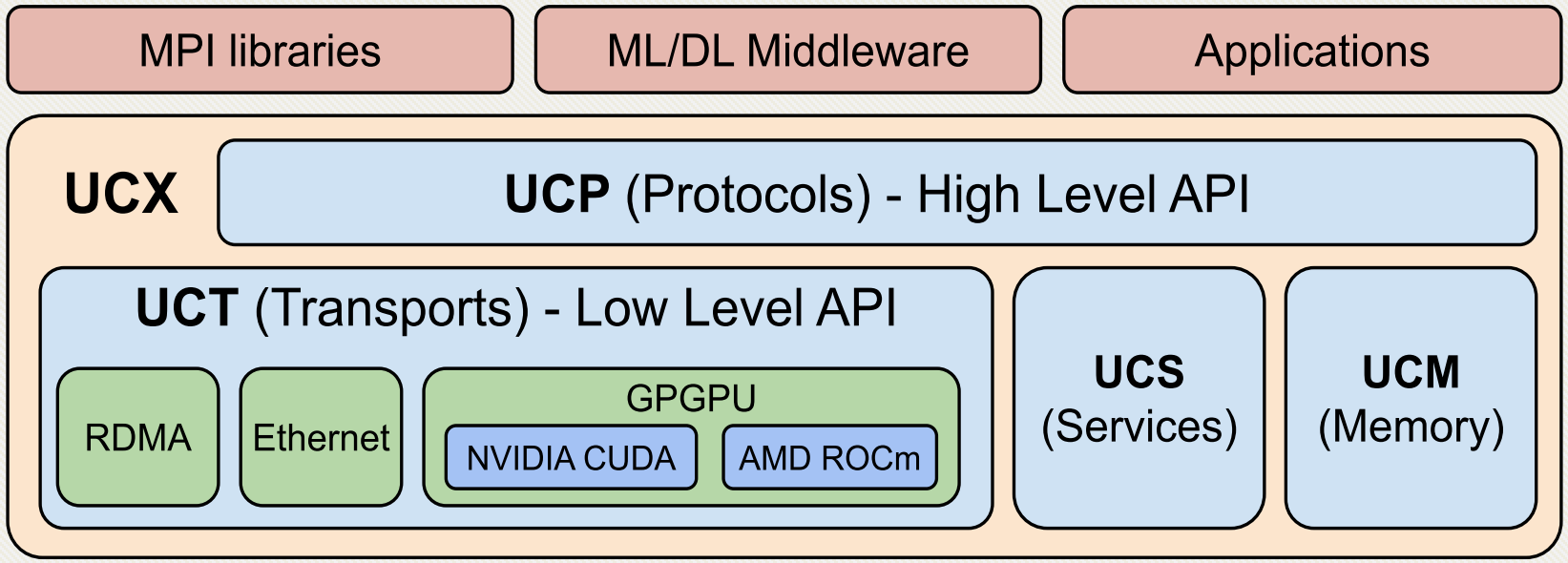


Figure 2: MPI and UCX high-level software stack

1. Multi-Path Communication Framework
2. Objectives
3. Strategies

Design Objectives

1. Multi-GPU Awareness
2. Path Selection
3. Communication Scheduling
4. Path Optimization
5. Data Integrity
6. Low Overhead

Framework Design

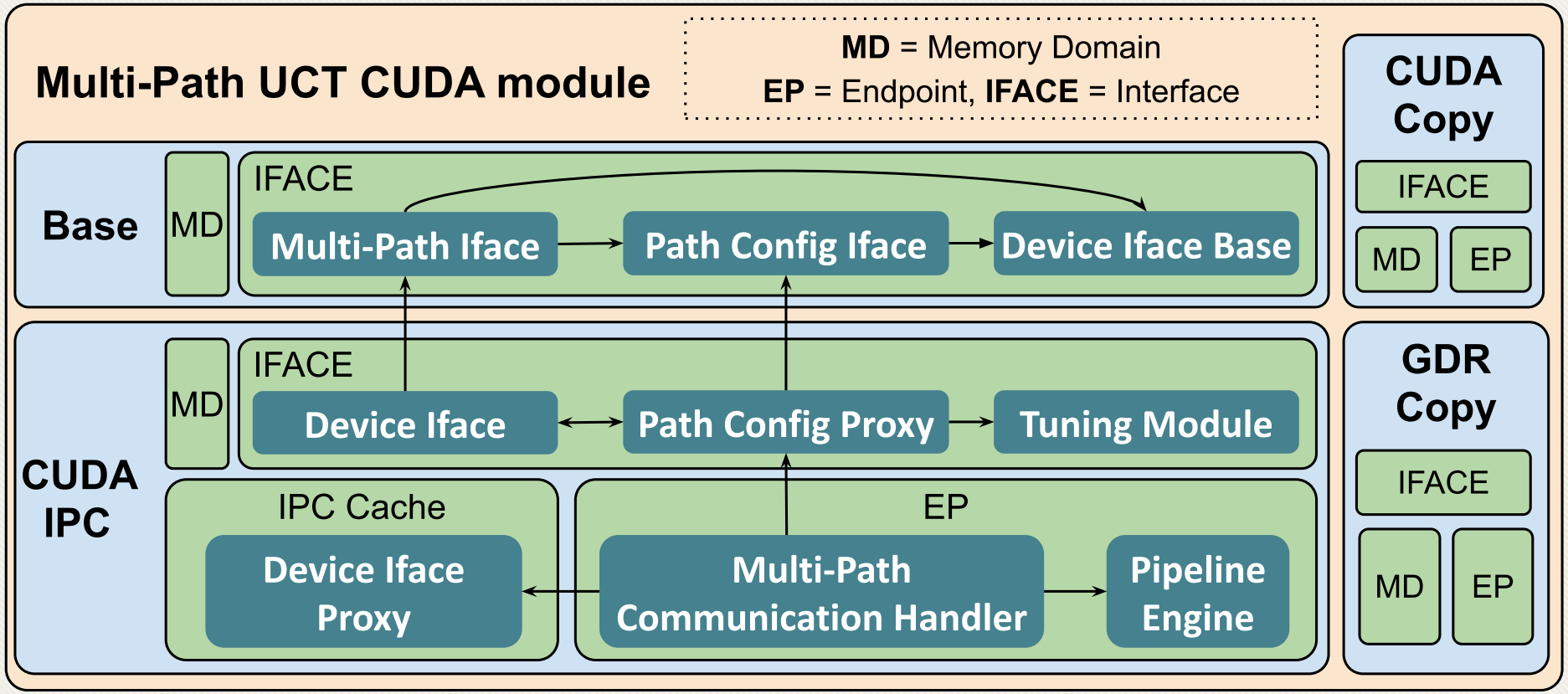
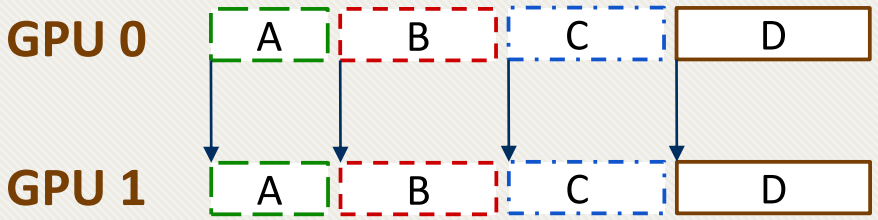


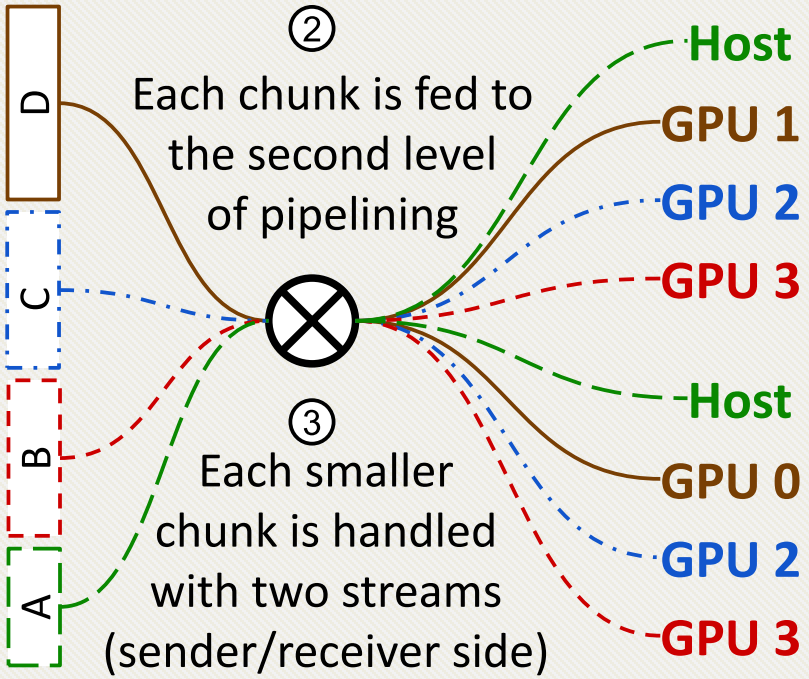
Figure 3: Framework Design integrated with UCT_CUDA module

2-D Pipeline Design

Message is dynamically split.

① Chunk sizes vary depending on the # of paths and tunings





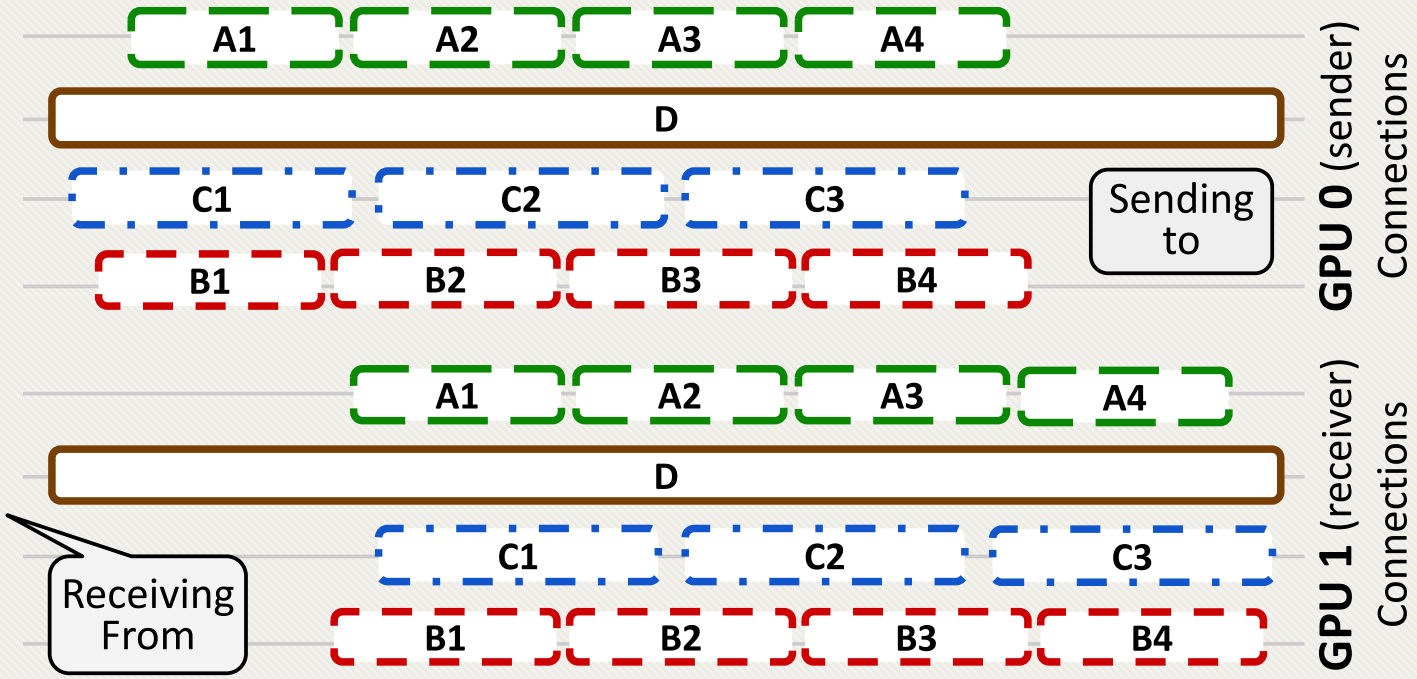


Figure 4: 2-D Pipeline in action

Ensuring Data Integrity

1. Data Corruption when staging
2. Data Dependency
3. Memory Ordering
4. Synchronization

- **A** Microbenchmarks – UCX Put Bandwidth
- **B** Microbenchmarks – OMB Bandwidth
- **C** Microbenchmarks – UCX Bidirectional Bandwidth
- **D** Application – Jacobi Iterative Solver

Experimental Setup

- Compute Canada's Beluga
  - 4 x NVIDIA V100 per node
  - Each GPU pair have two NVLinks
- Compute Canada's Narval
  - 4 x NVIDIA A100 per node
  - Each GPU pair have four NVLinks

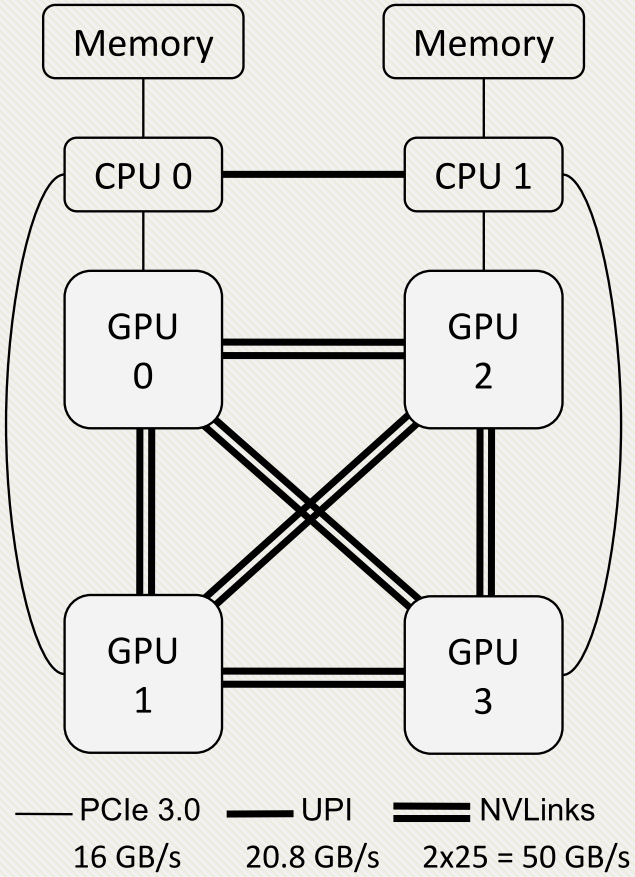


Figure 5: A typical Multi-GPU node

UCX Put Bandwidth

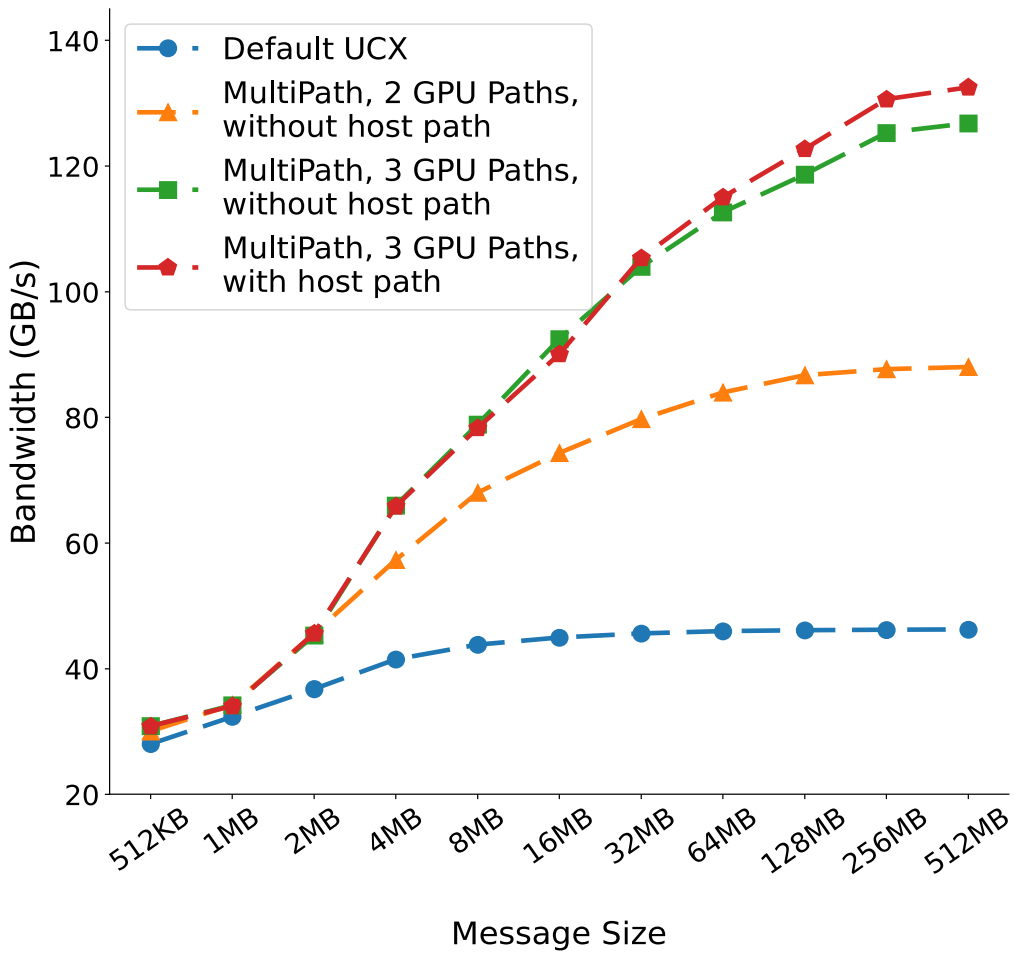


Figure 6.a : UCX Put Bandwidth Performance on Beluga

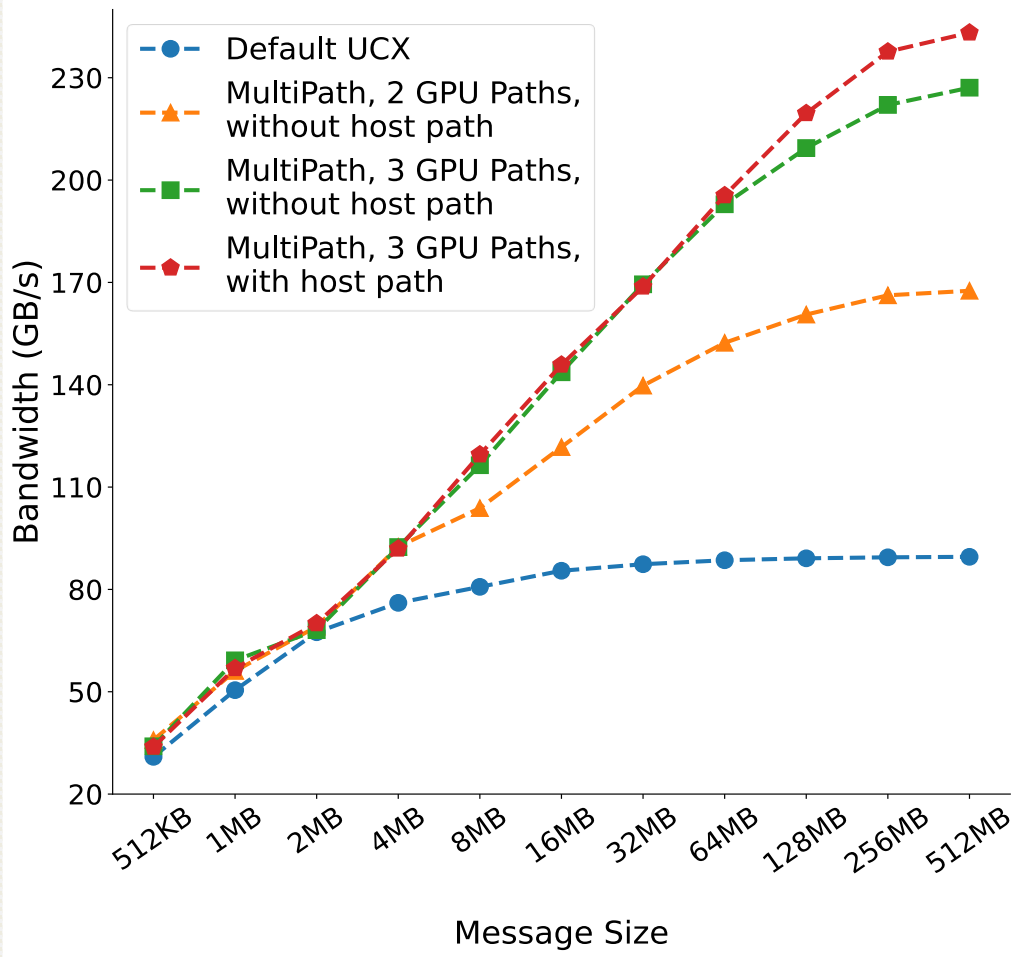


Figure 6.a : UCX Put Bandwidth Performance on Narval

OMB Bandwidth

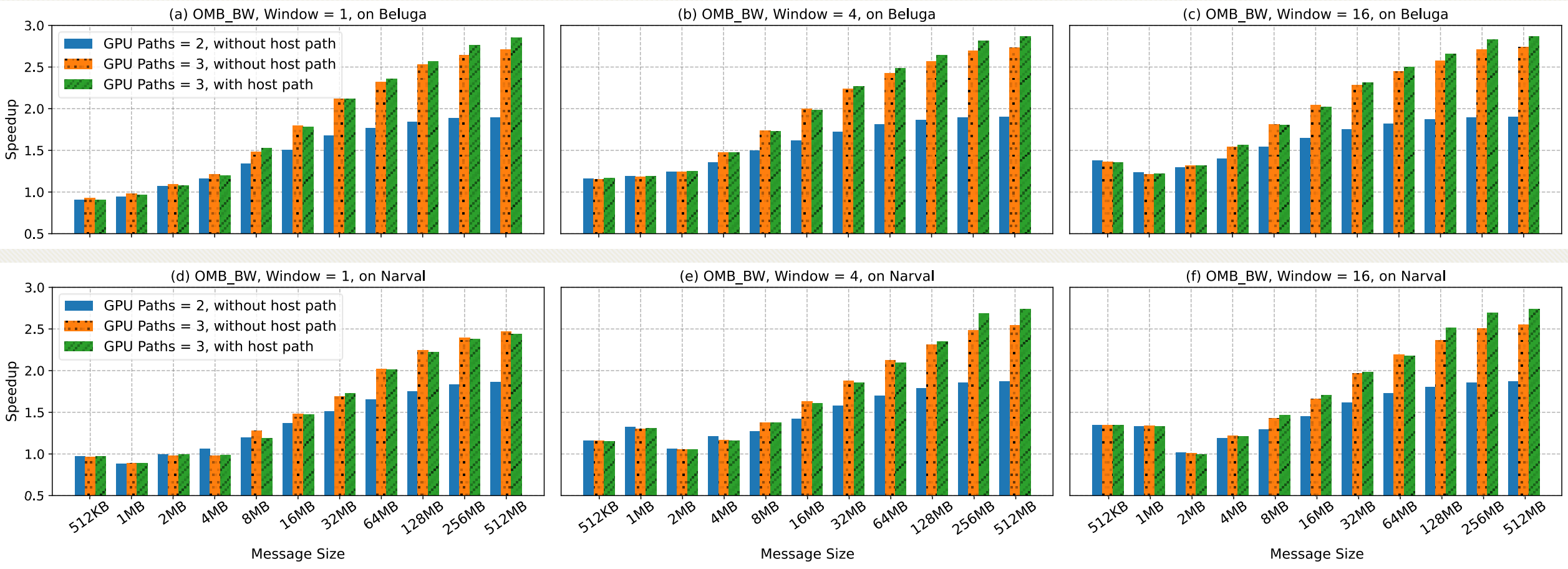


Figure7: OMB Bandwidth Performance Evaluation

OMB Bidirectional Bandwidth

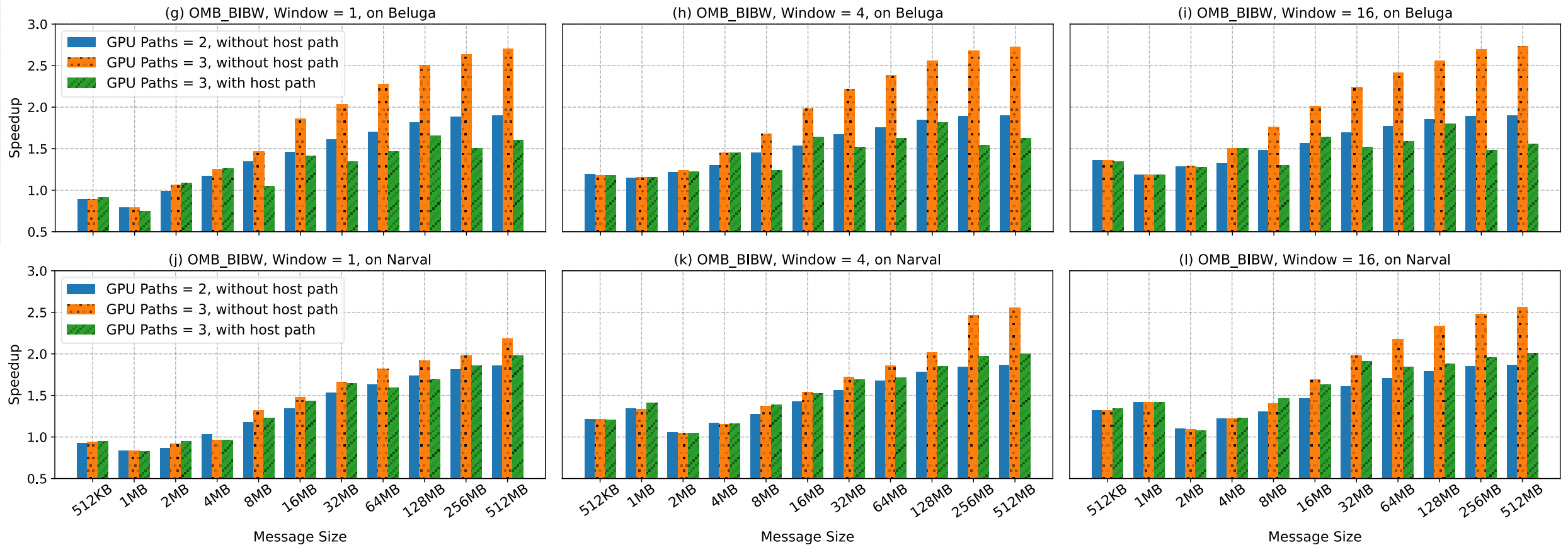


Figure8: OMB Bidirectional Bandwidth Performance Evaluation

Jacobi Iterative Solver – Communication Pattern

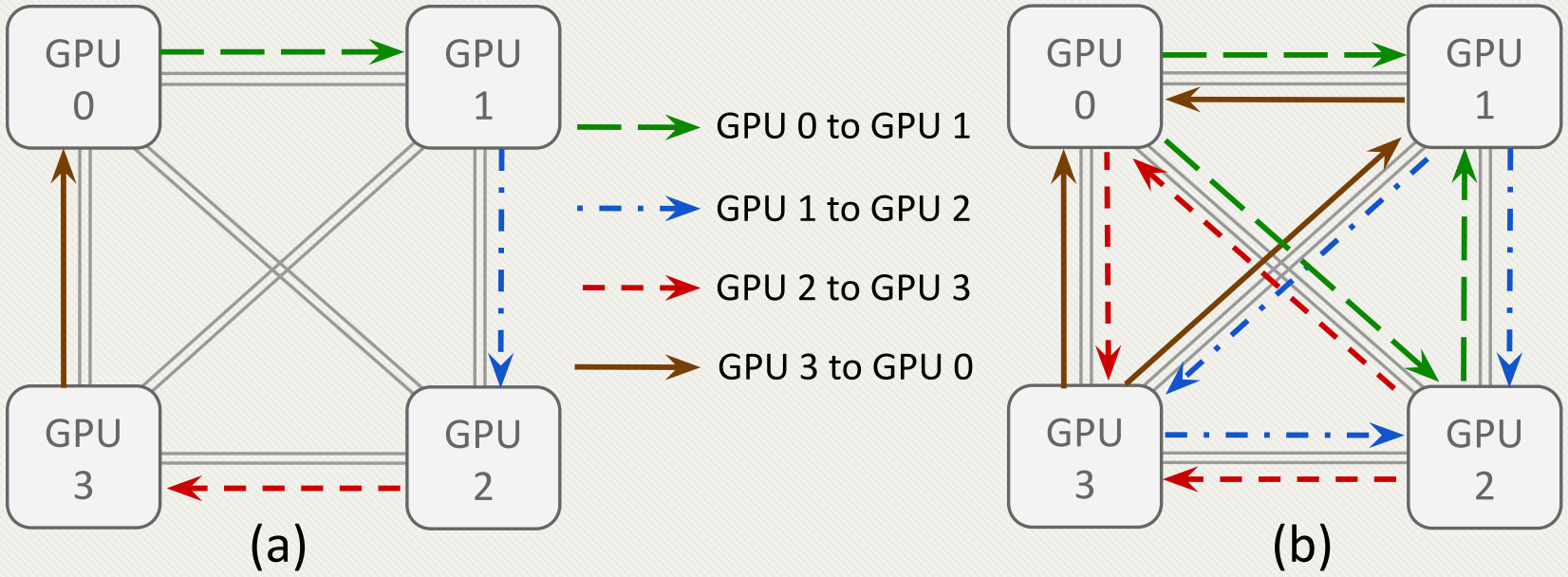


Figure 9: Jacobi Iterative Solver – Dual Path Communication

Jacobi Iterative Solver – Results

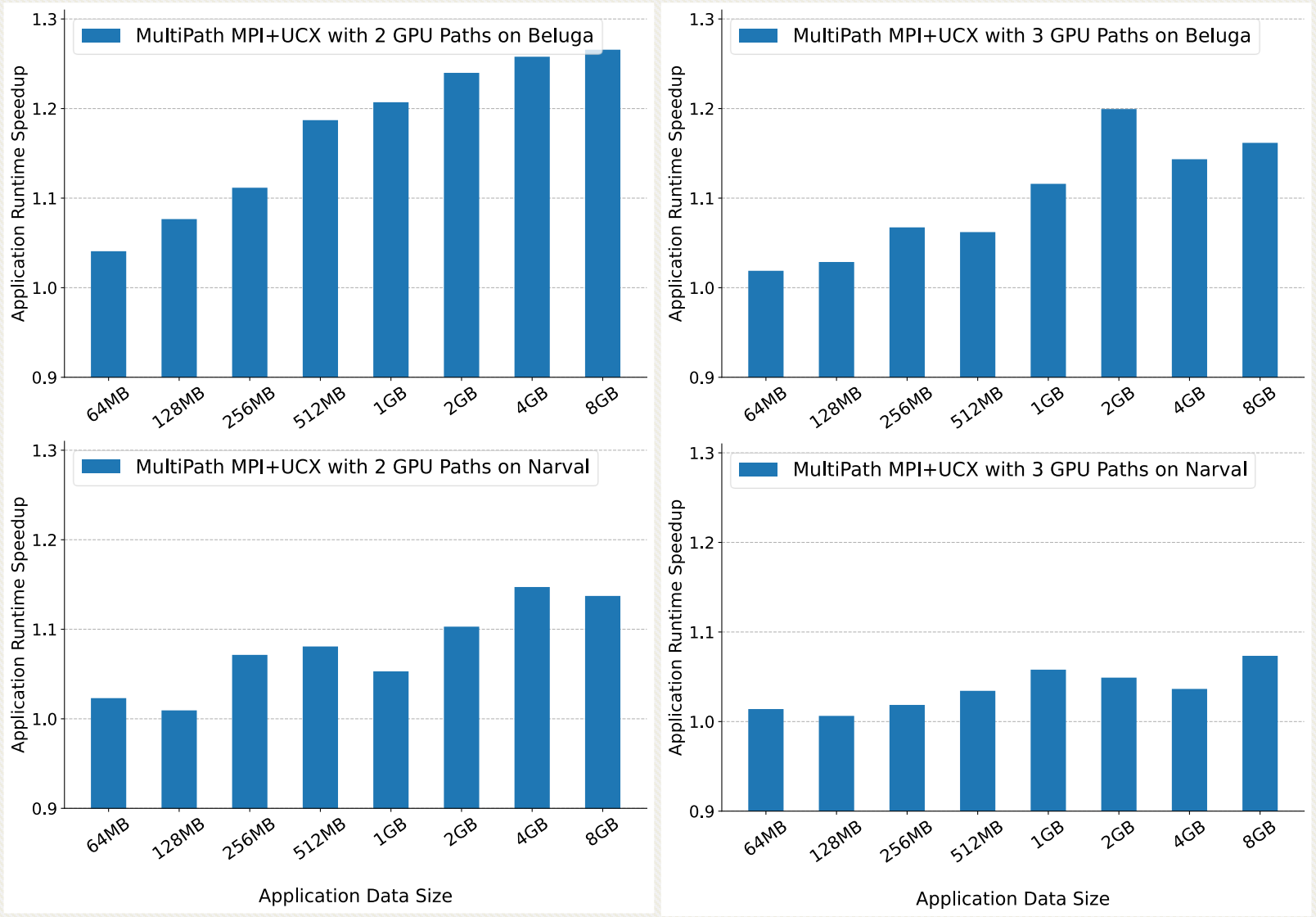


Figure 10: Jacobi Iterative Solver – Performance Evaluation

1. Conclusions
2. Future Directions

Conclusions

- Generalized Multi-Path within UCX
- Future Directions
  - Performance Modeling
  - Dynamic communication scheduling
    - Topology awareness
    - Automatically adapt to ongoing communication
    - Online tuning (e.g. through MPI_info objects)

Acknowledgments

- Natural Sciences and Engineering Research Council of Canada
- Digital Research Alliance of Canada

Thank You!

Instead of cursing the darkness, better light a candle!

Questions, Comments, and Ideas are Welcome!