Topology-Aware GPU Selection on Multi-GPU Nodes

Iman Faraji, Seyed H. Mirsadeghi, and Ahmad Afsahi

Department of Electrical and Computer Engineering
Parallel Processing Research Laboratory
Queen's University
Canada

The Sixth International Workshop on
Accelerators and Hybrid Exascale Systems (AsHES)

May 23, 2016

Outline

- Introduction
- Background and Motivation
- Design
- Results
- Conclusion
- Future Work

Outline

- **Introduction**
- Background and Motivation
- Design
- Results
- Conclusion
- Future Work

Introduction

- GPU accelerators have successfully established themselves in modern HPC clusters
  - High performance
  - Energy efficiency

- Demand for higher GPU computational power and memory
  - Multi-GPU nodes in state-of-the-art HPC clusters

Introduction

- Clusters with multi-GPU nodes provide:
  - ✓ Higher computational power
  - ✓ More memory to hold larger datasets

However, this brings up a challenge…

More GPUs → Potentially higher GPU-to-GPU communications

"Achilles heel" in GPU-accelerated application performance!

Introduction

- To address the GPU communication bottleneck:
  - Increase GPU utilization at the application level
    - ✓ Reducing the share of GPU communications in application runtime
    - ✗ Not all applications can highly utilize the GPUs in a node
  - Asynchronously progress inter-process GPU communications and GPU computation
    - ✓ Overlapping GPU communication with computation
    - ✗ Highly overlapping GPU communication and computation is not always feasible
  - Leverage GPU hardware features (such as IPC)
    - ✓ Improving GPU-to-GPU communication performance
    - ✗ Only possible for specific GPU pairs within a node
    - ✗ Communication performance still limited by the latency and bandwidth capacity

✗HOWEVER…

Introduction

- Smartly designed applications will continue to use these features
- GPU communications can still become a soft-point in different applications and GPU nodes

Conduct GPU communications **as efficient as possible**

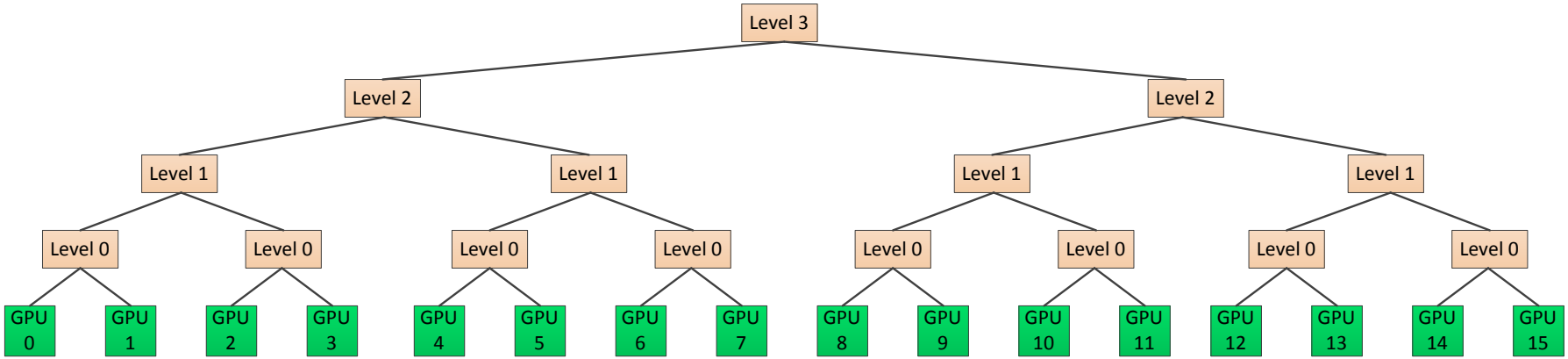
HOW?

Outline

- Introduction
- **Background and Motivation**
- Design
- Results
- Conclusion
- Future Work

Background and Motivation

- Multi-GPU node architecture

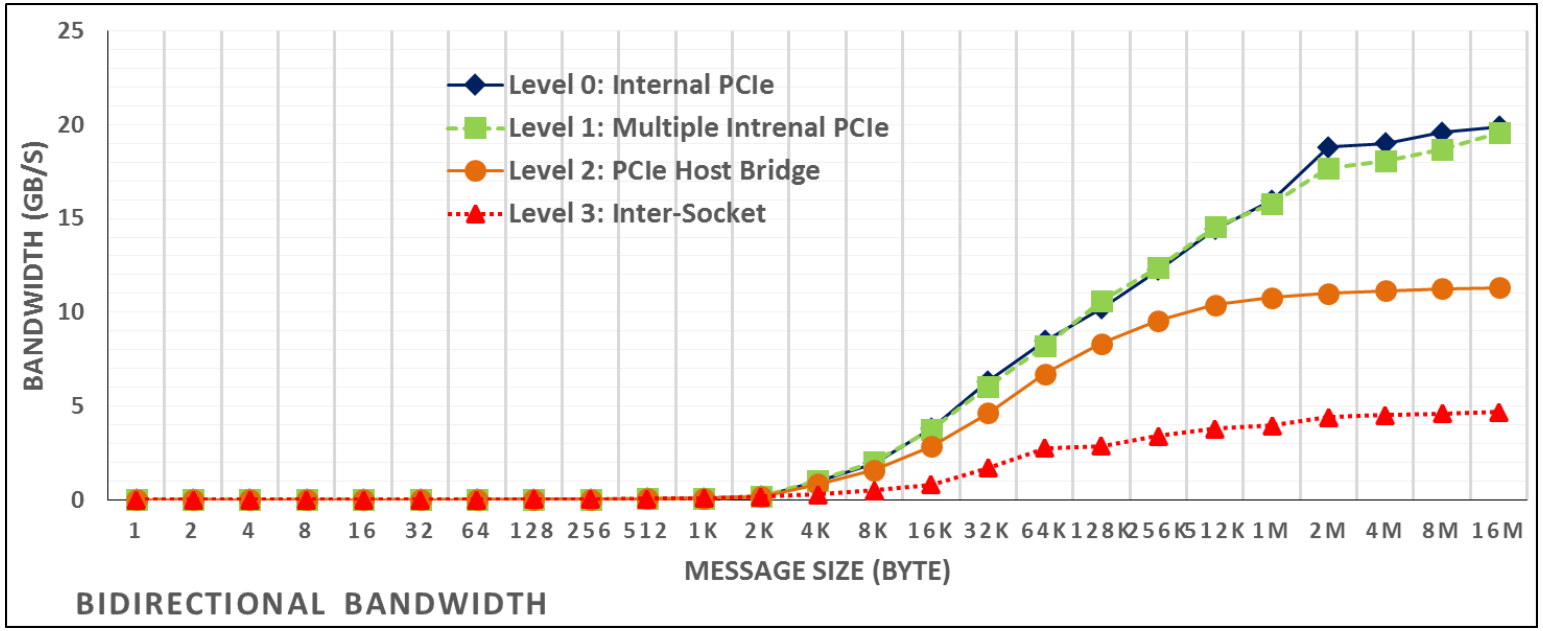


- **Level 0:** Path between GPU pairs traverses a PCIe internal switch
- **Level 1:** Path between GPU pairs traverses multiple internal switches
- **Level 2:** Path between GPU pairs traverses a PCIe host bridge
- **Level 3:** Path traverses a socket-level link (e.g., QPI)

Helios-K80 cluster at Université Laval's computing centre

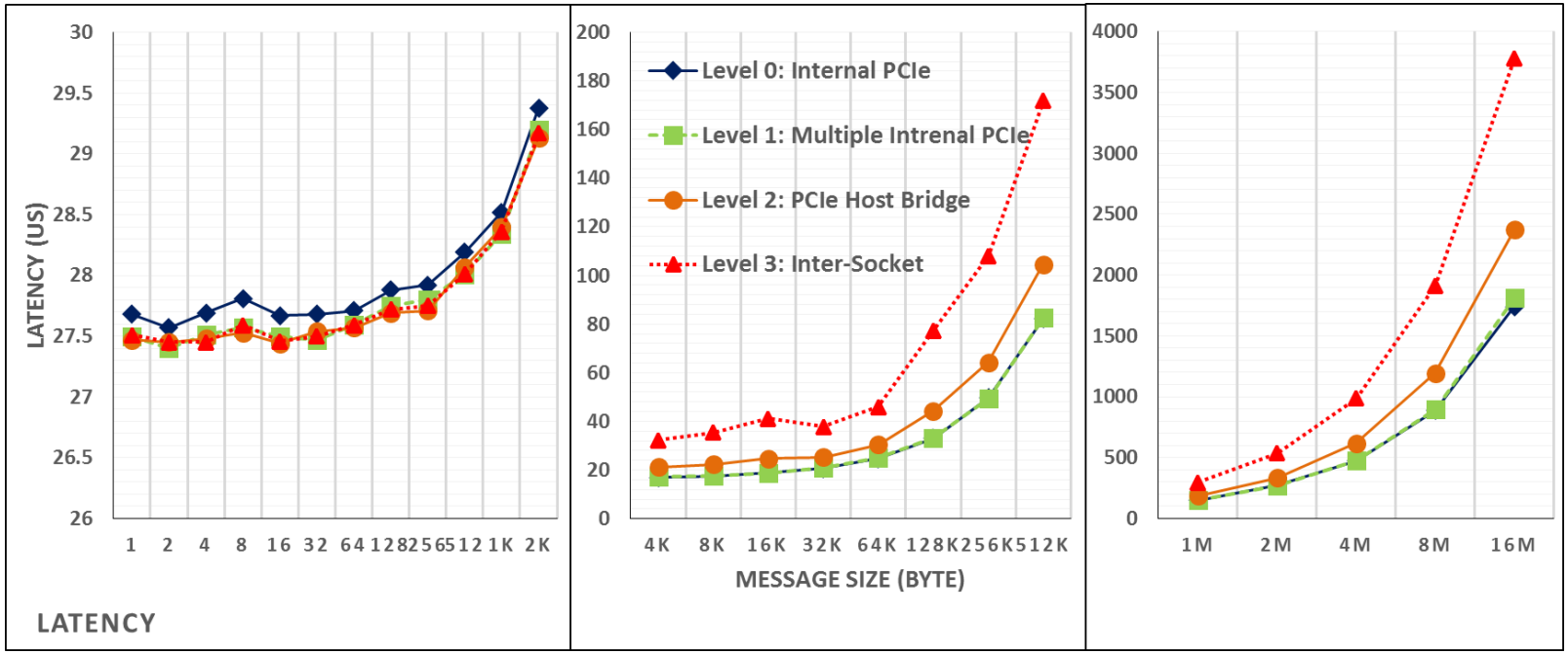
Background and Motivation

- Multi-GPU node bandwidth



Background and Motivation

- Multi-GPU node latency



Outline

- Introduction
- Background and Motivation
- **Design**
- Results
- Conclusion
- Future Work

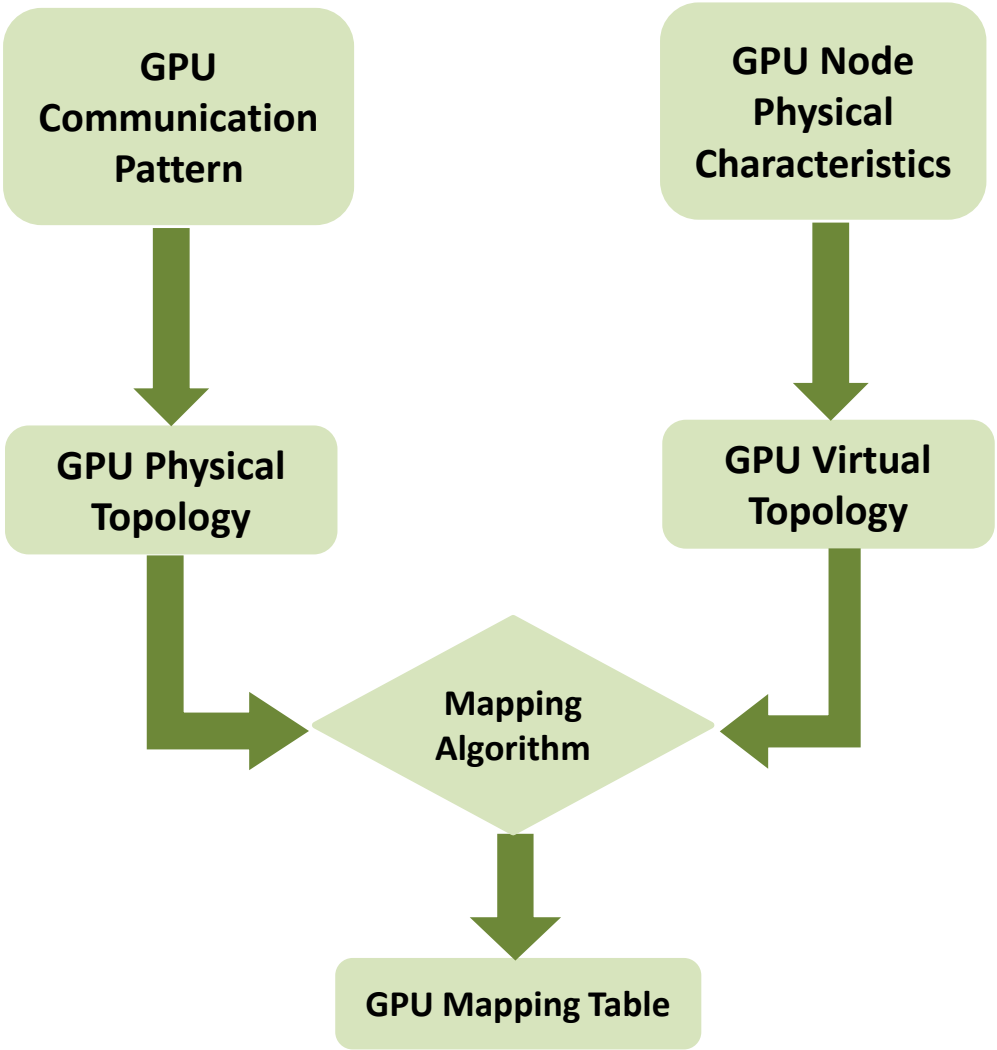
Design

- What we know:
  - Intranode GPU-to-GPU communications may traverse different paths
  - Different paths can have different latency and bandwidth
- Ultimate goal:
  - Efficient utilization of GPU communication channels
    - Intensive communications carried over stronger channels
- Our proposal:
  - **Topology-aware GPU selection**
    - Intelligent assignment of Intranode GPUs to MPI processes so as to maximize communication performance

Design

Our Approach:

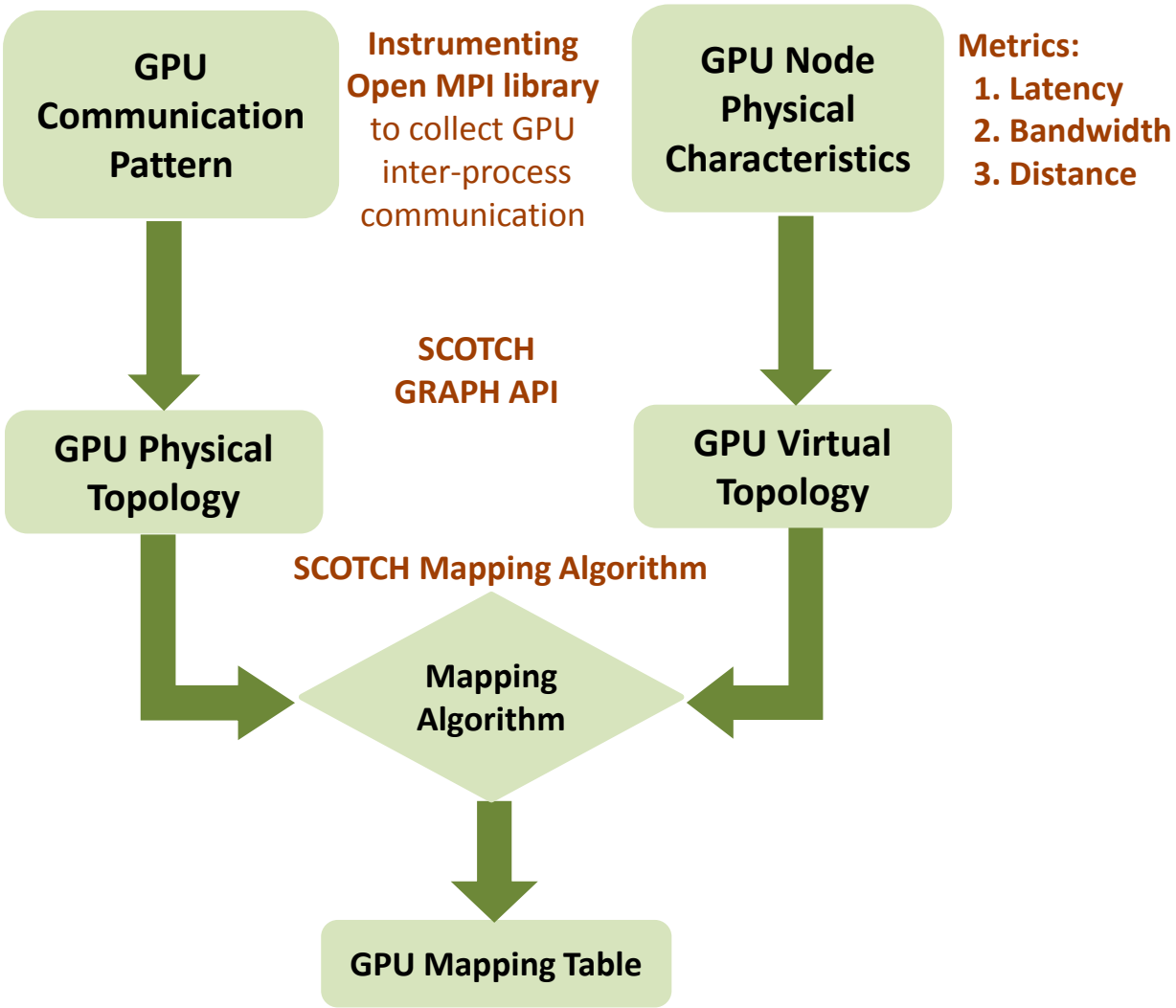
1. Extracting the GPU communication pattern
2. Extracting the physical characteristics of the node
3. Modeling topology-aware GPU selection as a graph mapping problem
4. Solving the problem using a mapping algorithm



Design

Our Approach:

1. Extracting the GPU communication pattern
2. Extracting the physical characteristics of the node
3. Modeling topology-aware GPU selection as a graph mapping problem
4. Solving the problem using a mapping algorithm

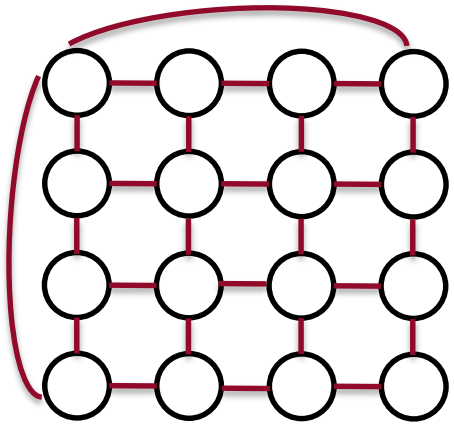


Outline

- Introduction
- Background and Motivation
- Design
- **Results**
- Conclusion
- Future Work

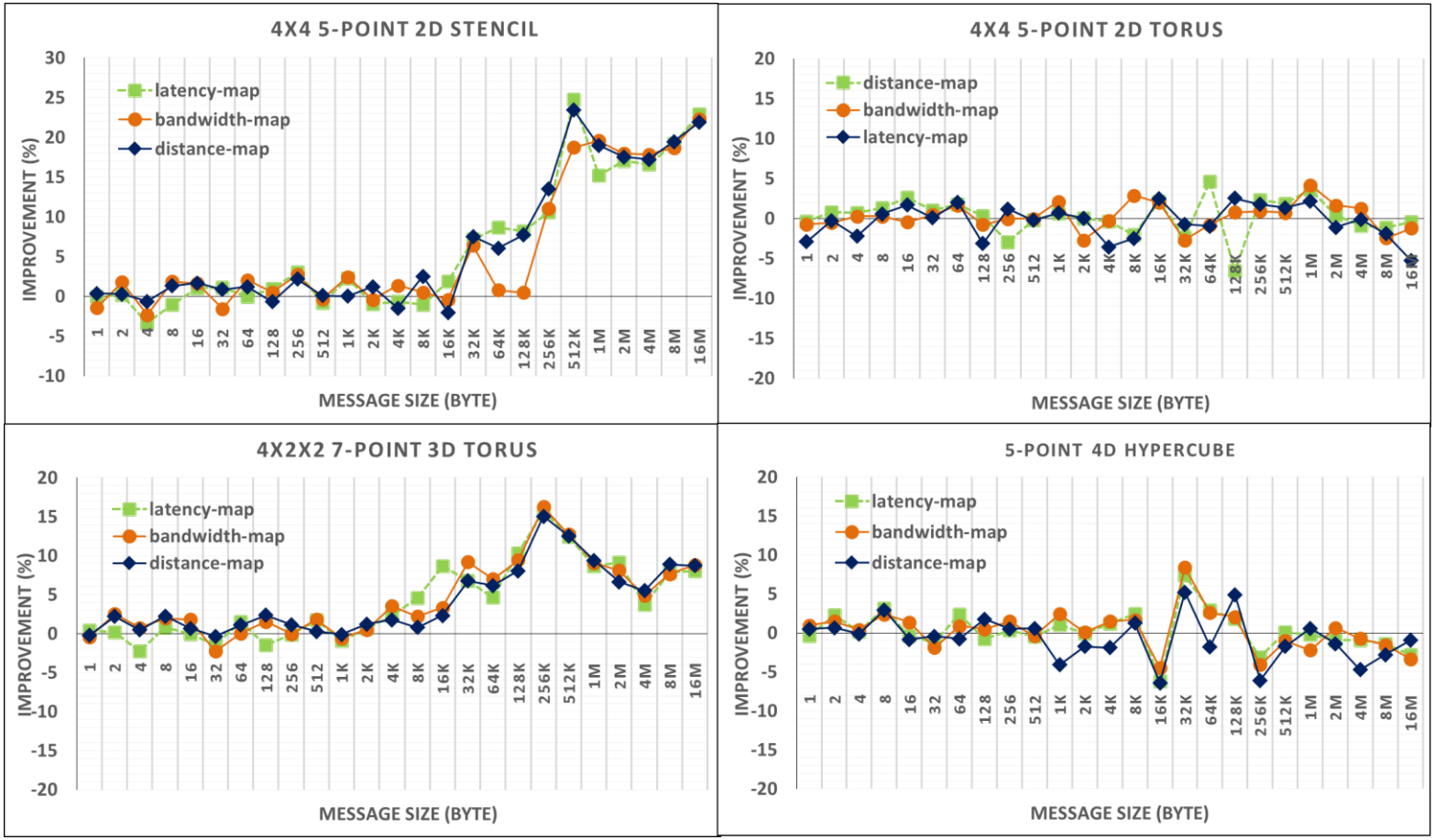
Result: Setup

- One node, Helios cluster from Calcul Quebec
  - 16 GPUs (K80)
  - Two 12-core Intel Xeon 2.7 GHz
- 4 micro-benchmarks
  - 5-point 2D stencil
  - 5-point 2D torus
  - 7-point 3D torus
  - 5-point 4D hypercube
- One application (New)



Results

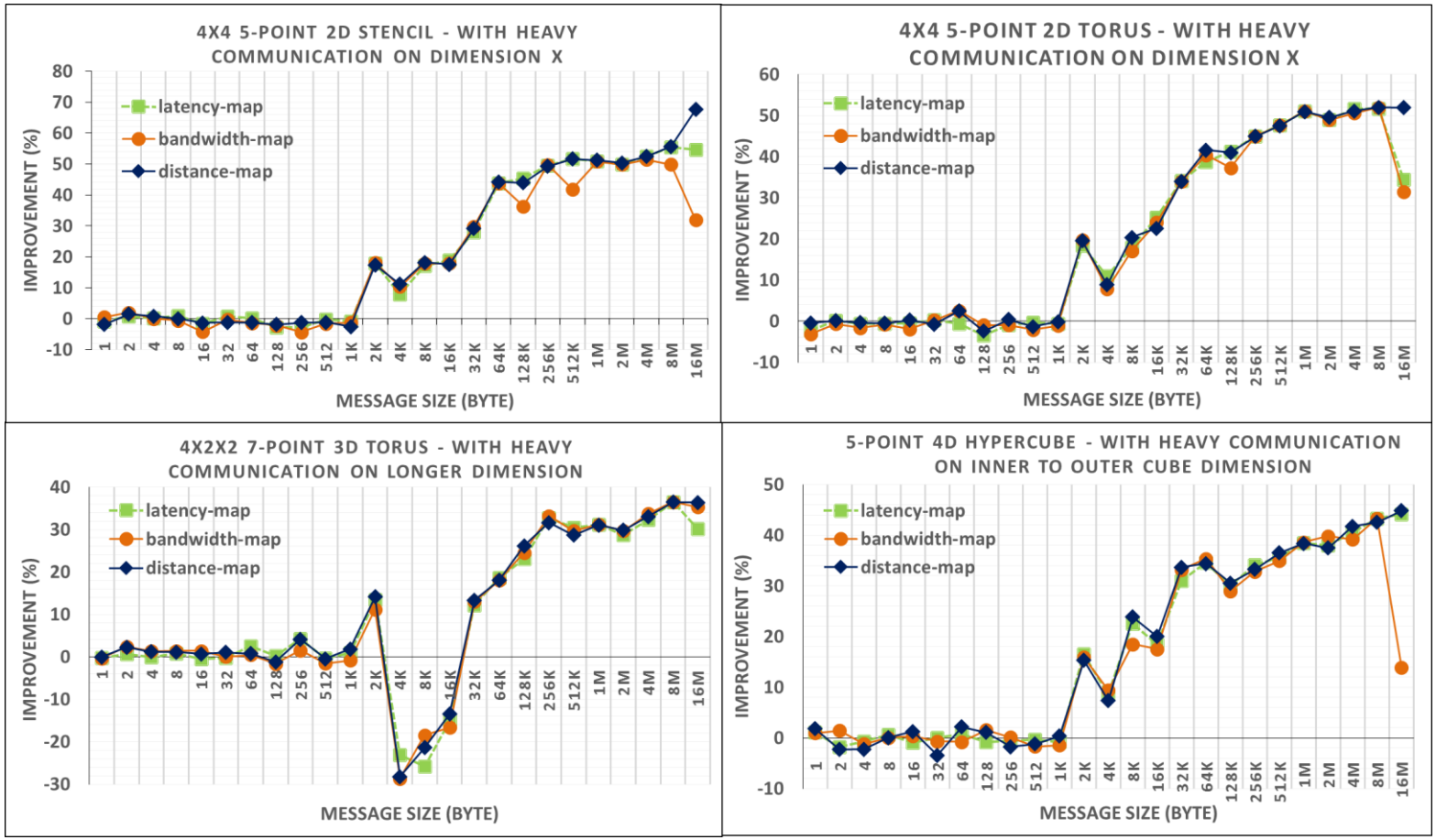
Micro-benchmark



Runtime improvement of topology-aware mappings over default mapping on **non-weighted microbenchmarks**

Results

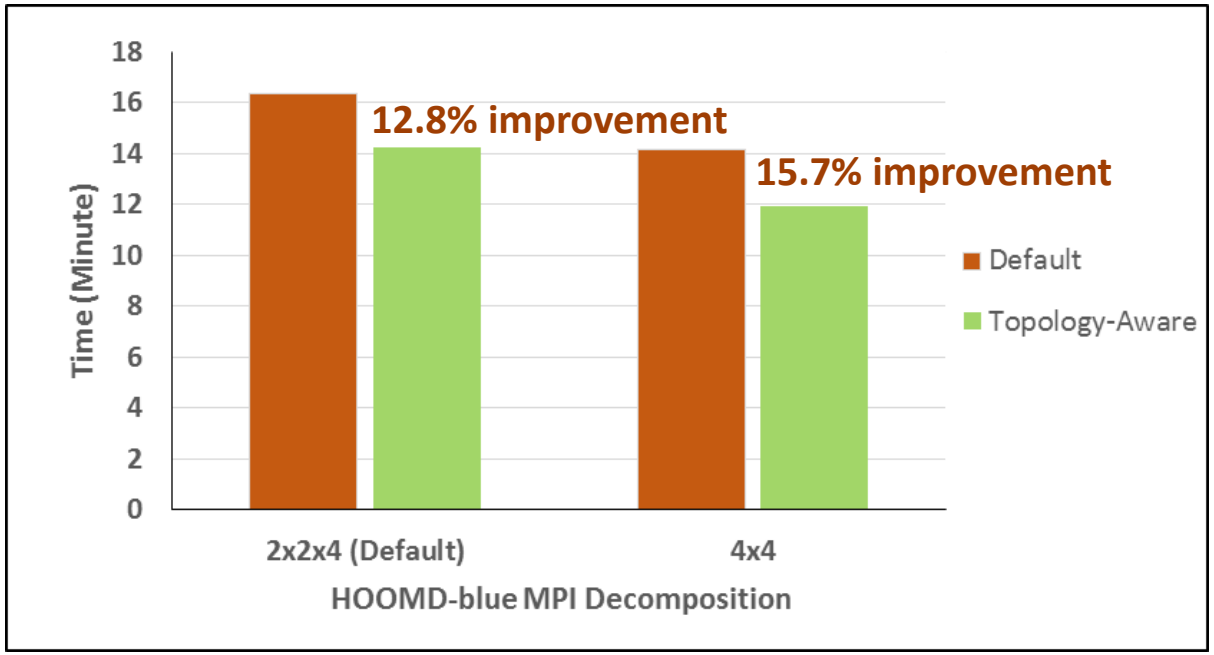
Micro-benchmark



Runtime improvement of topology-aware mappings over default mapping on **weighted microbenchmarks**

NEW!

Results
Application



Runtime of the **HOOMD-Blue application with LJ-512K particle size** using default and topology-aware mappings

Conclusion

- Discussed GPU inter-process communication bottleneck
  - Overviewed some potential solutions to subside its effect
- Showed an example of a multi-GPU node and its communication channels
- Showed different levels of bandwidth and latency in a Multi-GPU node
- Proposed a topology-aware GPU selection approach
  - More efficient utilization of GPU-to-GPU communication channels
  - Performance improvement by mapping intensive communications onto stronger channels

Conclusion

Topology awareness matters for GPU communications

and

can provide considerable performance improvements.

Future Work

- Evaluation on different multi-GPU nodes with different node architectures and GPUs.
- Impact on different applications.
- Extension towards multiple nodes across the cluster.

Acknowledgments

Queen's University

NSERC CRSNG

Canada Foundation for Innovation
Fondation canadienne pour l'innovation

compute canada | calcul canada

Ontario Innovation Trust

Calcul Québec

Thank you for your attention!

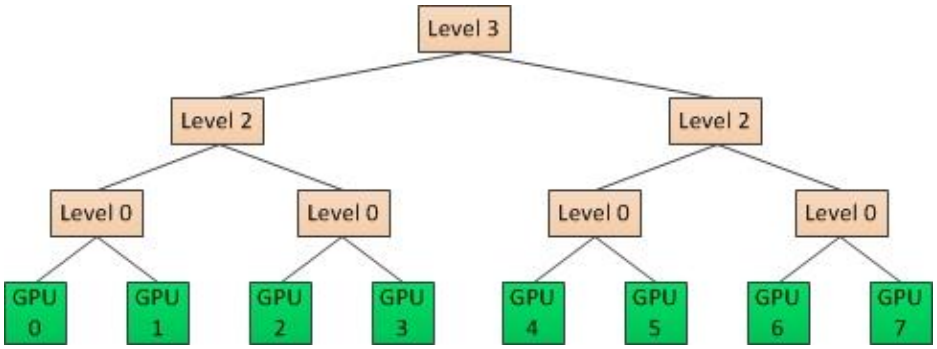
Contacts:

- **Iman Faraji**: i.faraji@queensu.ca
- **Seyed H. Mirsadeghi**: s.mirsadeghi@queensu.ca
- **Ahmad Afsahi**: ahmad.afsahi@queensu.ca

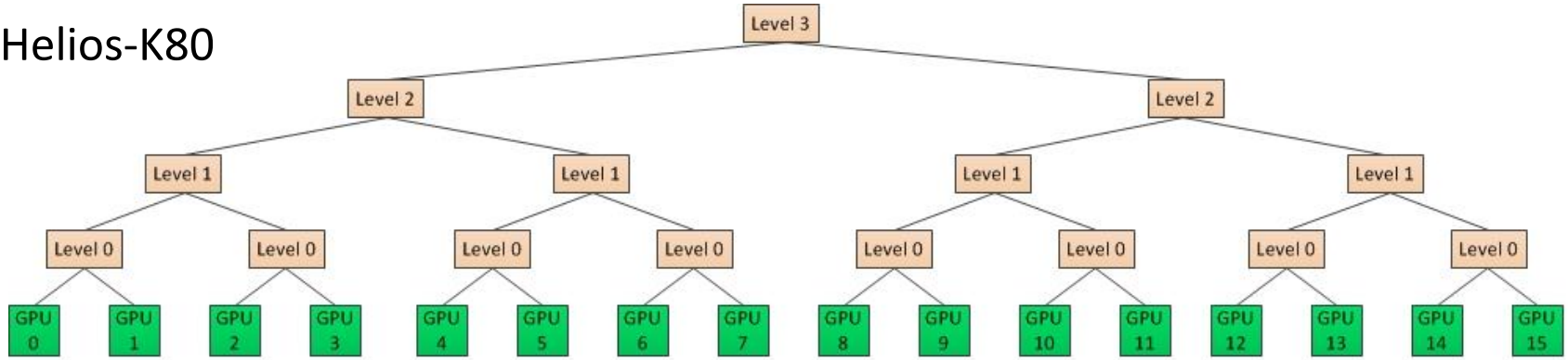
Question?

Backup
Motivation

Helios-K20



Helios-K80



- **Level 0:** Path between GPU pairs traverses a PCIe internal switch
- **Level 2:** Path between GPU pairs traverses a PCIe host bridge
- **Level 1:** Path between GPU pairs traverses multiple internal switches
- **Level 3:** Path traverses a socket-level link (e.g., QPI)