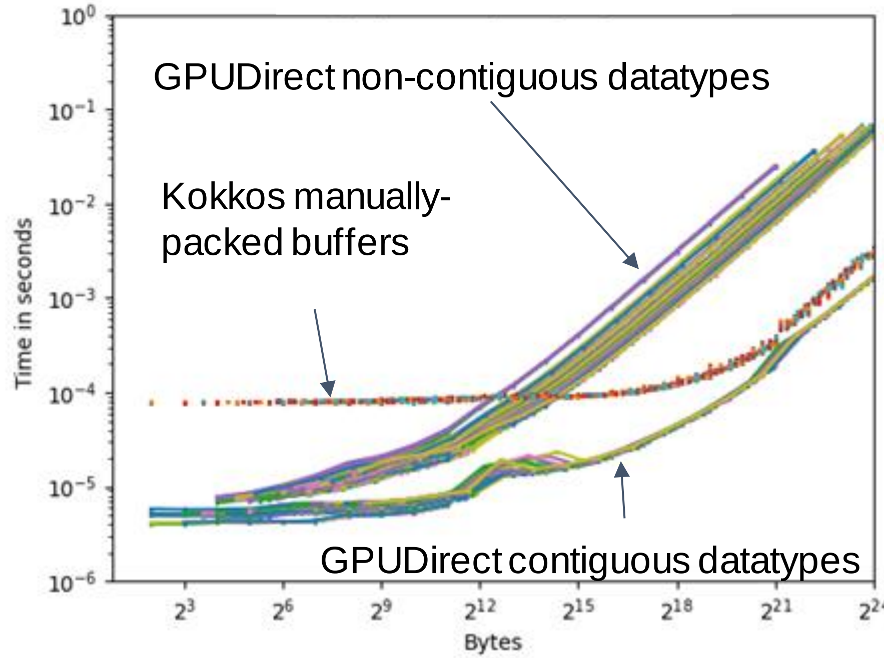
Evaluating the Viability of LogGP for Modeling MPI Performance with Non-contiguous Datatypes on Modern Architectures

Nicholas Bacon[*], Patrick G. Bridges[*], Scott Levy[^],
Kurt Ferreira[*^], and Amanda Bienz[*]

*Department of Computer Science, University of New Mexico
^Center for Computing Research, Sandia National Laboratories

Can LogP models help us better understand datatype performance?

- Datatypes are an essential element of MPI to describe complex buffer layouts
- Datatype performance challenging on modern GPU systems
- Datatype performance varies significantly, and be difficult to understand and predict



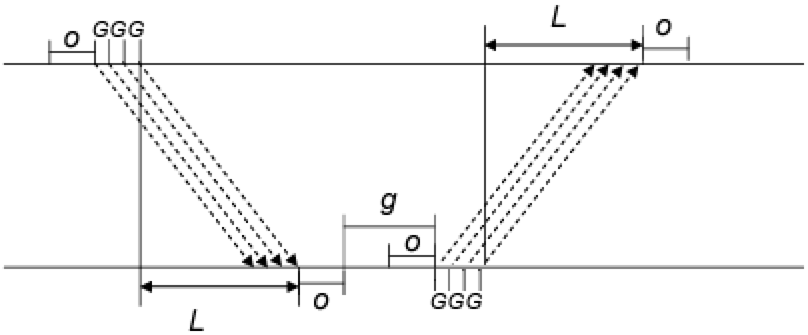
MVAPICH ping pong latency of different MPI datatypes on LLNL Lassen

Contributions

- Analysis of suitability of LogGOP-based models to quantify modern MPI communication performance
  - GPU-based systems
  - Non-contiguous data
- Modified open-source NetGauge tool for measuring LogGOP parameters on GPU systems
- Evaluation of LogGOP accuracy on GPU systems and with non-contiguous data
- Model-based comparison of MPI implementations and HPC systems handling non-contiguous data

LogP family of network models

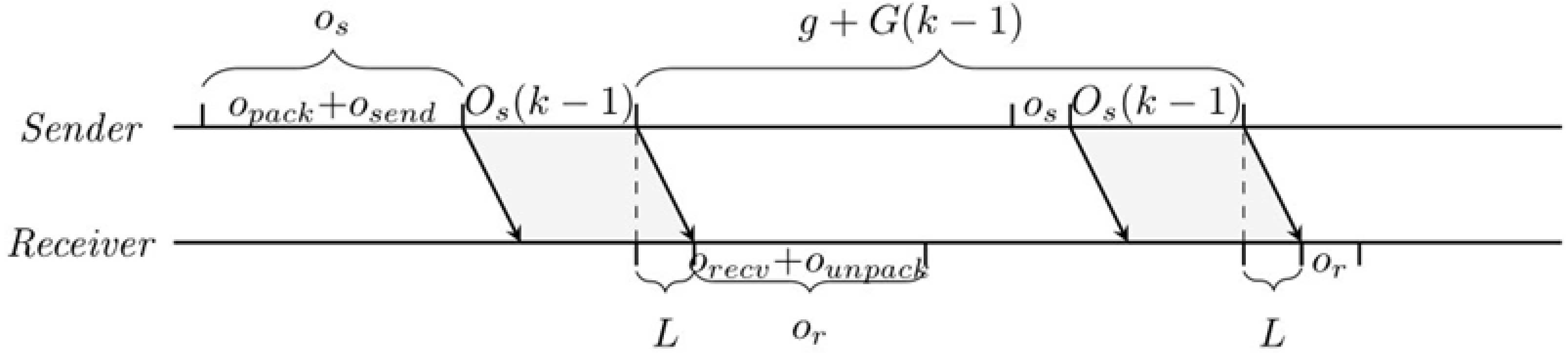
- Straight-forward parameterization of the network communication
- LogGP parameters
  - L = latency
  - o = overhead
  - g = gap
  - G = gap per byte
  - P = cost per byte
- LogGOP parameters
  - Decompose original o
  - Per-message overhead (o)
  - Per-byte overhead (O)



LogGP representation of a ping pong data exchange

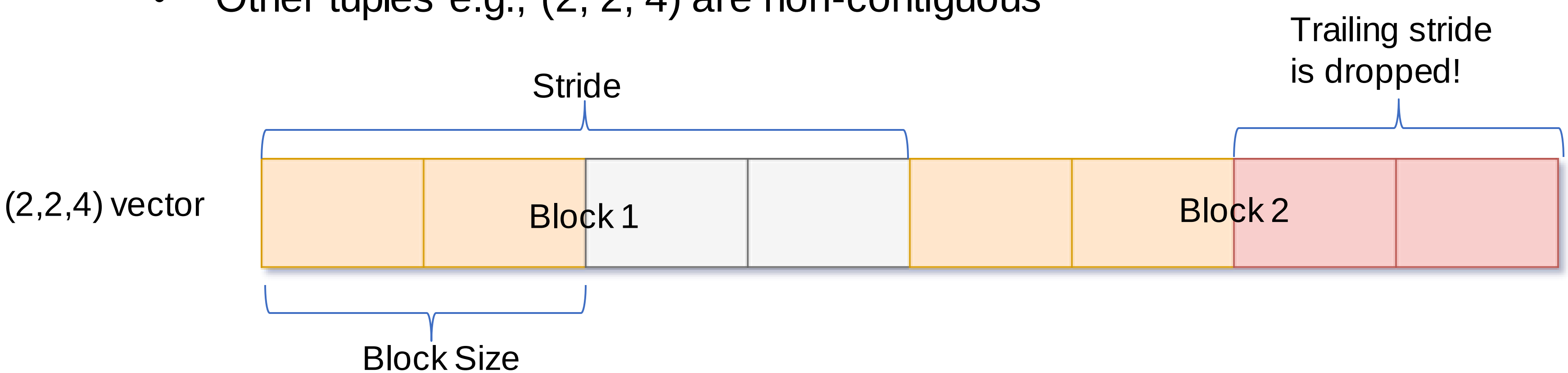
Mapping LogGOP to GPU communication systems

- GPU-related communication costs modeled as overhead
  - Data packing and unpacking
  - Copying data between host and GPU memory
- Model packing and unpacking as part of LogGOP overhead
  - O now includes latency for packing (o_{pack}) and unpacking (o_{unpack})
  - o now includes bandwidths for packing and unpacking



What Datatypes to measure?

- Focused on modeling and measurement of MPI_Type_vector – simplest non-primitive
- Varied (block count, block size, stride) tuple to include both contiguous and non-contiguous datatype
- Selected stride of 4, block counts and sizes strides from 1-4 (details in paper)
- Reminder
  - Block count of 1 e.g., (1, X, Y) is contiguous (trailing stride is dropped)
  - Block size = block stride e.g., (2, 4, 4) is contiguous.
  - Other tuples e.g., (2, 2, 4) are non-contiguous



Modifying NetGauge for GPUs and non-contiguous data

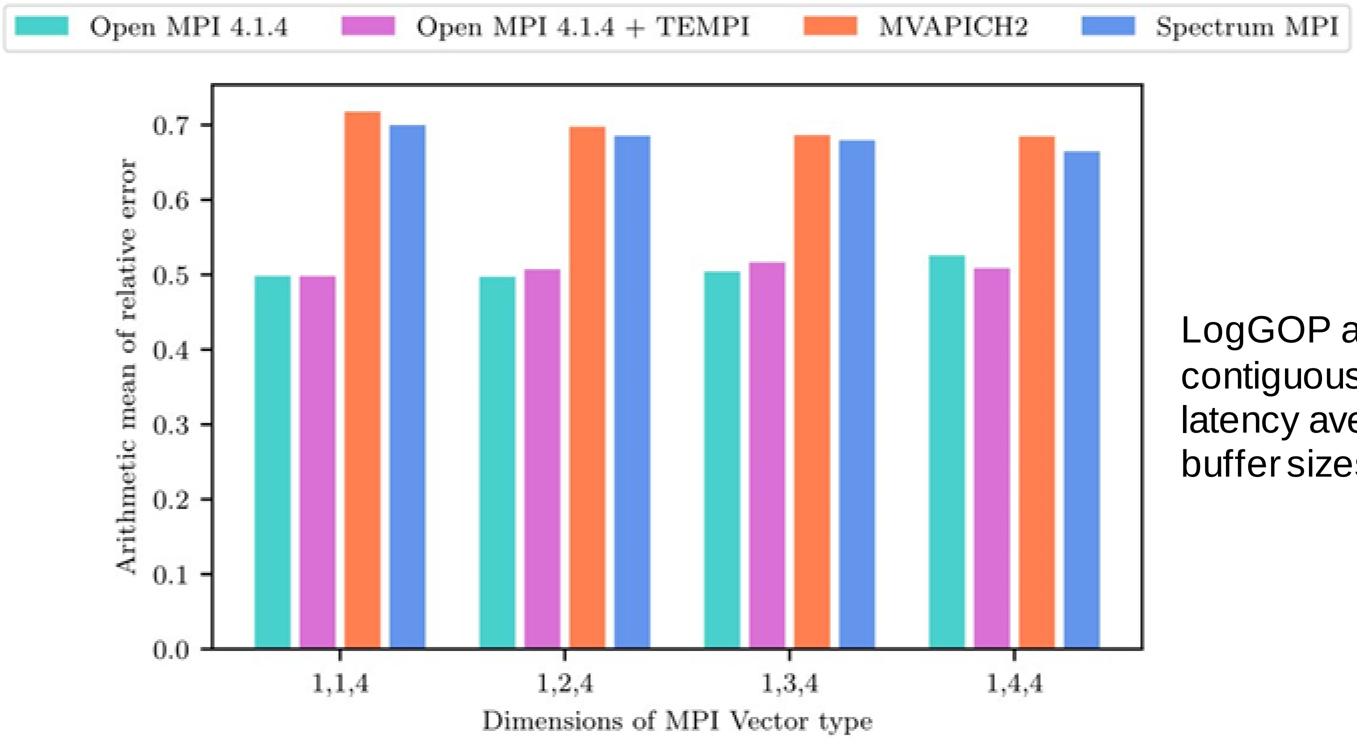
- Add support for MPI Vector datatypes
- Enable usage of GPU memory for data buffers.
- Increased RTT parameter to exceed observed maximum round trip latency with GPU datatypes on Lassen
- Available as open source (URL in paper)

Methodology

- Use modified NetGauge to model ping pong performance on different systems and MPIs
- Compare against median ping-pong latency
- Systems and MPI Implementations Tested
  - Lassen: IBM POWER9 CPUs, NVIDIA V100, IB HDR
    - Spectrum MPI module version 2020.08.19.
    - MVAPICH2-GDR 2021.05.29 with Cuda/11.1.1.
  - Glinda: AMD EPYC CPUs, NVIDIA A100, IB HDR
    - OpenMPI4 4.1.4
    - OpenMPI4+TEMPI: Include TEMPI datatype engine.

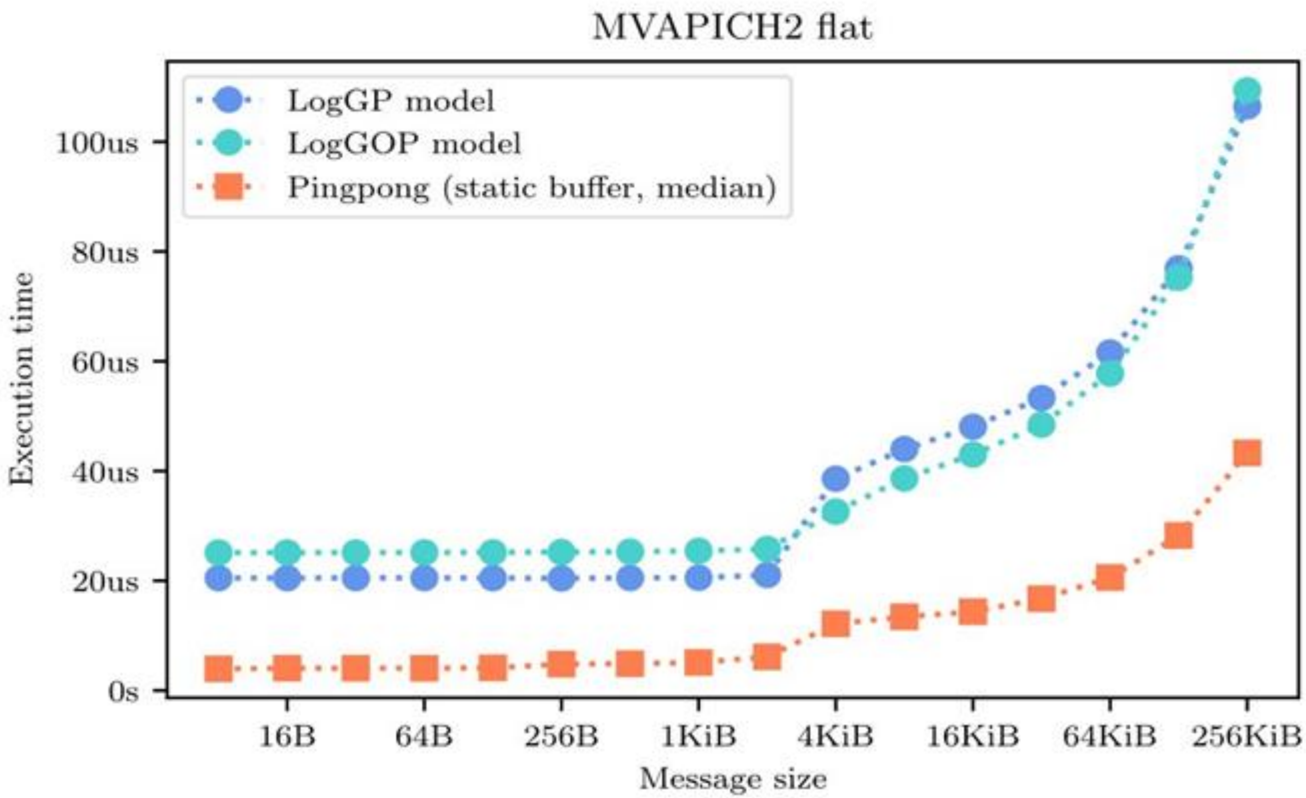
How accurate are LogGOP and LogGP for contiguous buffers?

Absolute communication performance prediction accuracy poor



LogGOP accuracy with contiguous datatype ping pong latency averaged across all buffer sizes

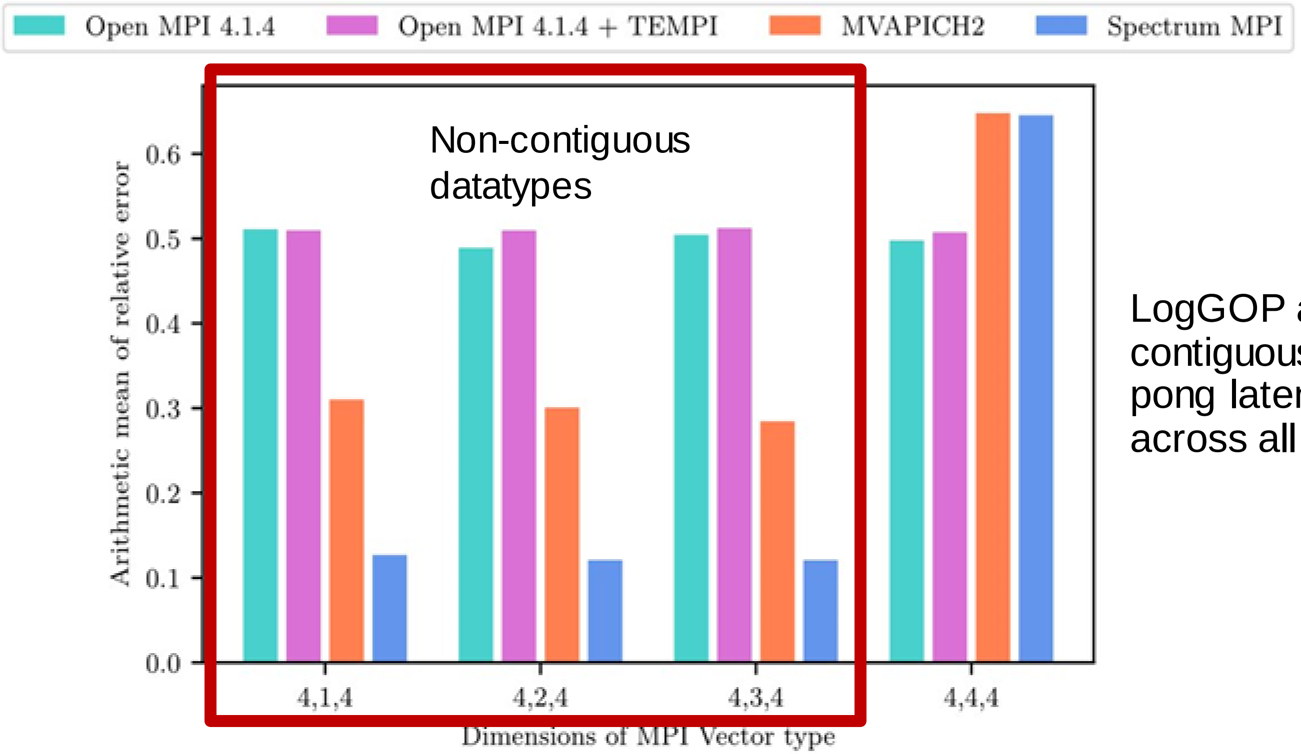
Model still captures general communication trends



- LogGP and LogGOP modeled performance versus measured ping pong latency with a flat buffer
- Primitive MPI_FLOAT datatype) on MVAPICH2 on Lassen
- Similar performance with contiguous datatypes

How accurate are LogGOP and LogGP for non-contiguous buffers?

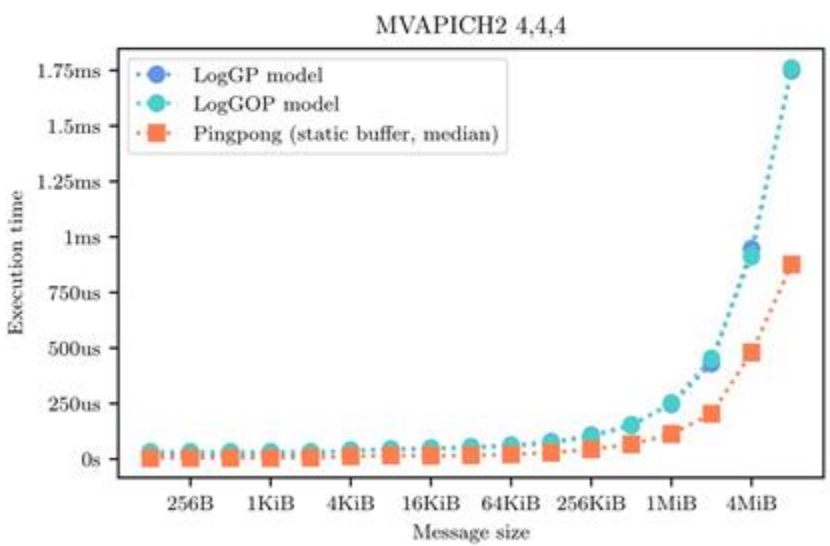
Better performance prediction accuracy with non-contiguous buffers



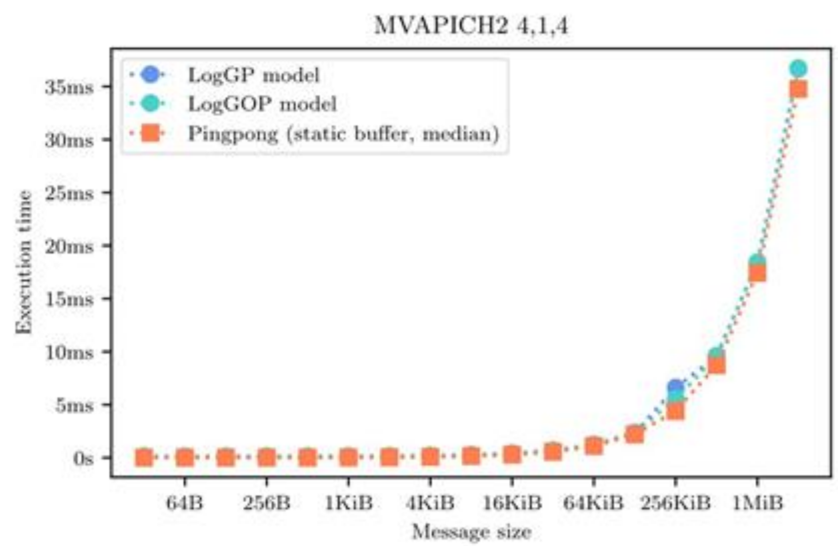
LogGOP accuracy with contiguous datatype ping pong latency averaged across all buffer sizes

Better model accuracy with non-contiguous datatypes

Model captures datatype packing and unpacking overheads better than communication costs

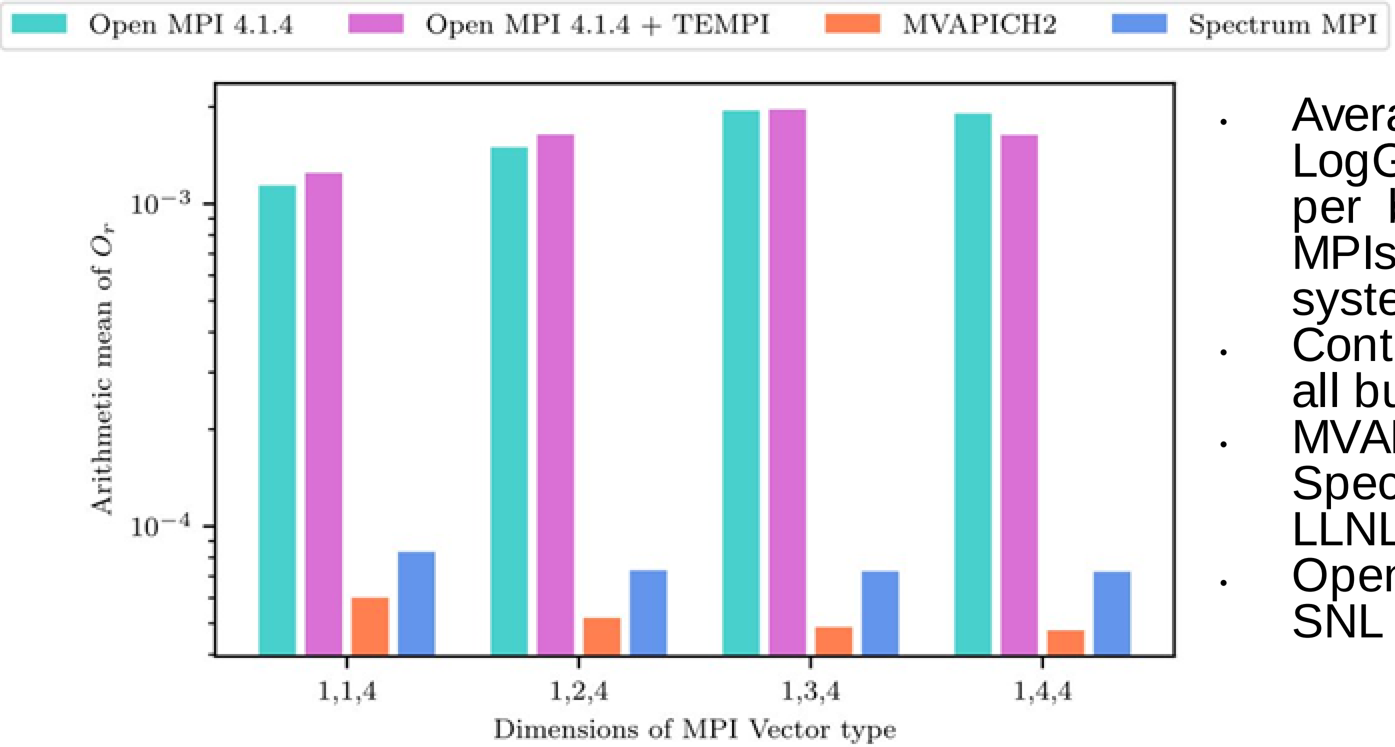


Contiguous



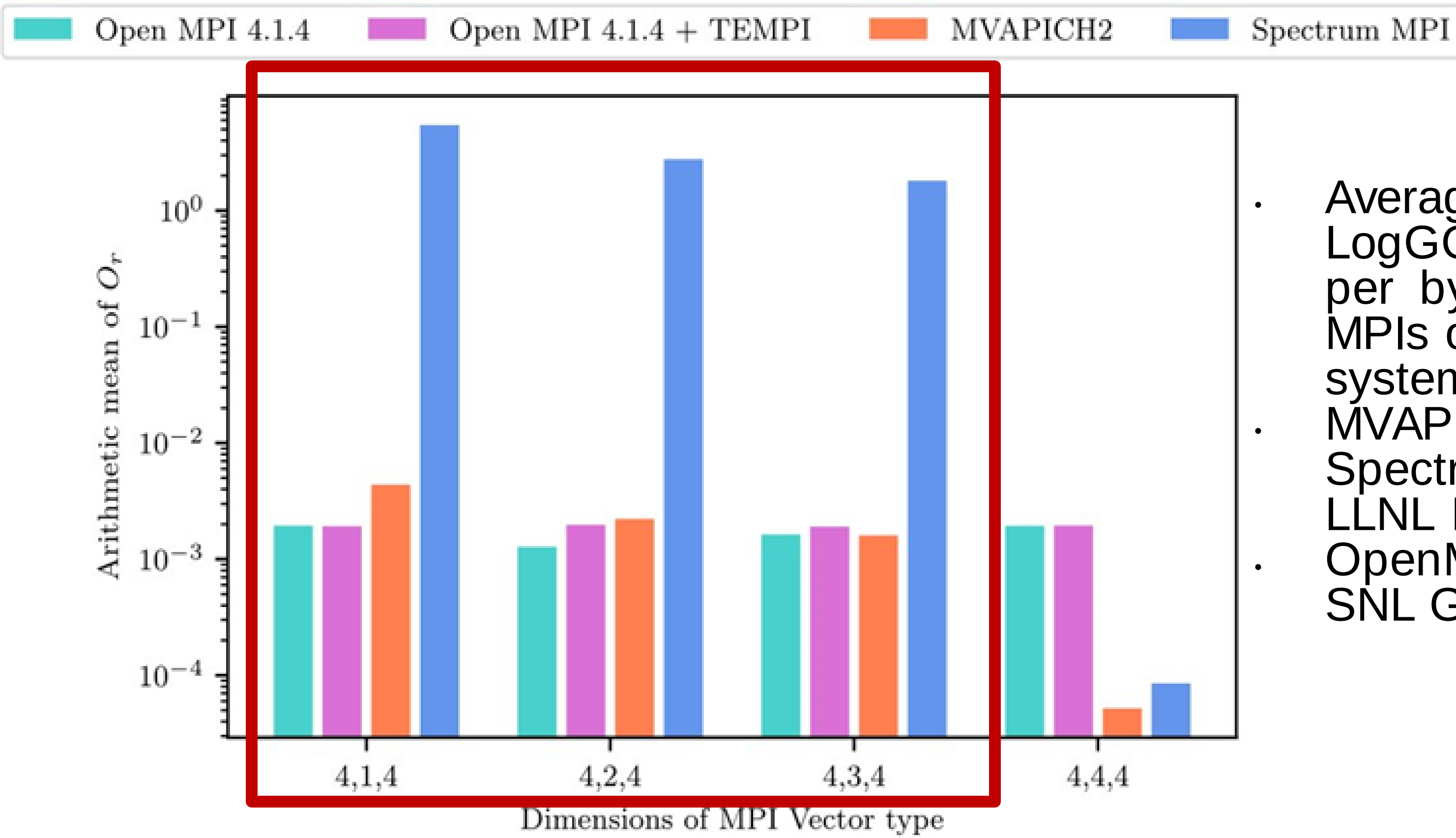
Non-contiguous

Model quantifies datatype overheads in different MPI implementations



- Average modeled LogGOP overhead per byte of different MPIs on different systems
- Contiguous datatype all buffer sizes
- MVAPICH and Spectrum from LLNL Lassen
- OpenMPI from SNL Glinda

Overheads per byte generally higher with non-contiguous buffers



- Average modeled LogGOP overhead per byte of different MPIs on different systems
- MVAPICH and Spectrum from LLNL Lassen
- OpenMPI runs from SNL Glinda

Summary of Results

1. The LogGP and LogGOP models generally tracks the trends of measured communication performance
   a. Overestimates ping-pong times for primitive and derived datatypes.
   b. Tend to over-predict ping-pong communication times, especially for very large and very small messages.
2. The LogGP and LogGOP models can effectively quantify the performance of communication using MPI derived datatypes
   a. communication using more expensive sparse datatypes where datatype packing/unpacking costs dominate network communication costs.
3. The LogGP and LogGOP models can be used to quantify the performance of contiguous and non-contiguous communication data

Acknowledgements

- National Science Foundation OAC-2103510
- U.S. Dept. of Energy Award DE-NA0003966
- Sandia National Laboratories