KOÇ UNIVERSITY

parCorelab

ComScribe: Identifying Intra-node GPU communication

Palwisha Akhtar, Erhan Tezcan, Fareed M. Qararyah and Didem Unat

Koç University, Istanbul, Turkey

Motivation

- GPUs are increasingly becoming common and vital in compute-intensive applications
  - With single GPU and exceeding working sets, memory becomes bottleneck
    - Results in a trend towards multi-GPU computing
- Variety of single node Multi-GPU systems
  - Varying number of GPUs
  - Different Topologies
  - Device Capabilities

Motivation

2 GPUs (TU102)

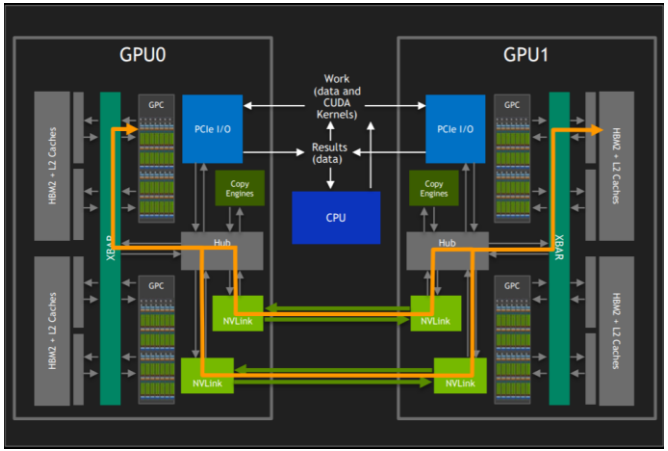


Image source: https://www.hotchips.org/hc30/2conf/2.01_Nvidia_NVswitch_HotChips2018_DGX2NVS_Final.pdf

8 GPUs (DGX-1)

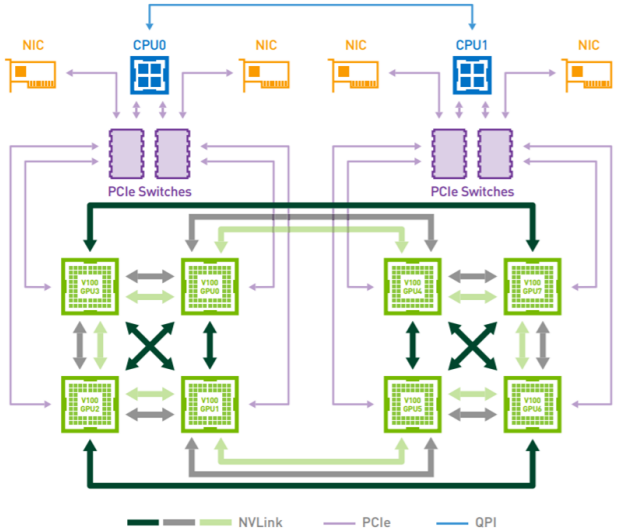


Image source: http://images.nvidia.com/content/pdf/dgx1-v100-system-architecture-whitepaper.pdf

16 GPUs (DGX-2)

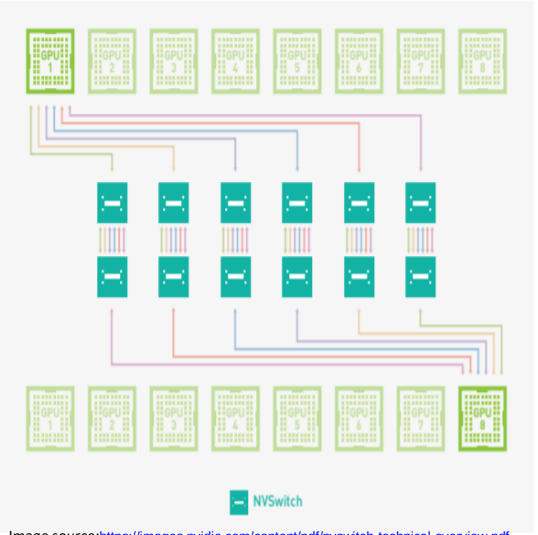


Image source:https://images.nvidia.com/content/pdf/nvswitch-technical-overview.pdf

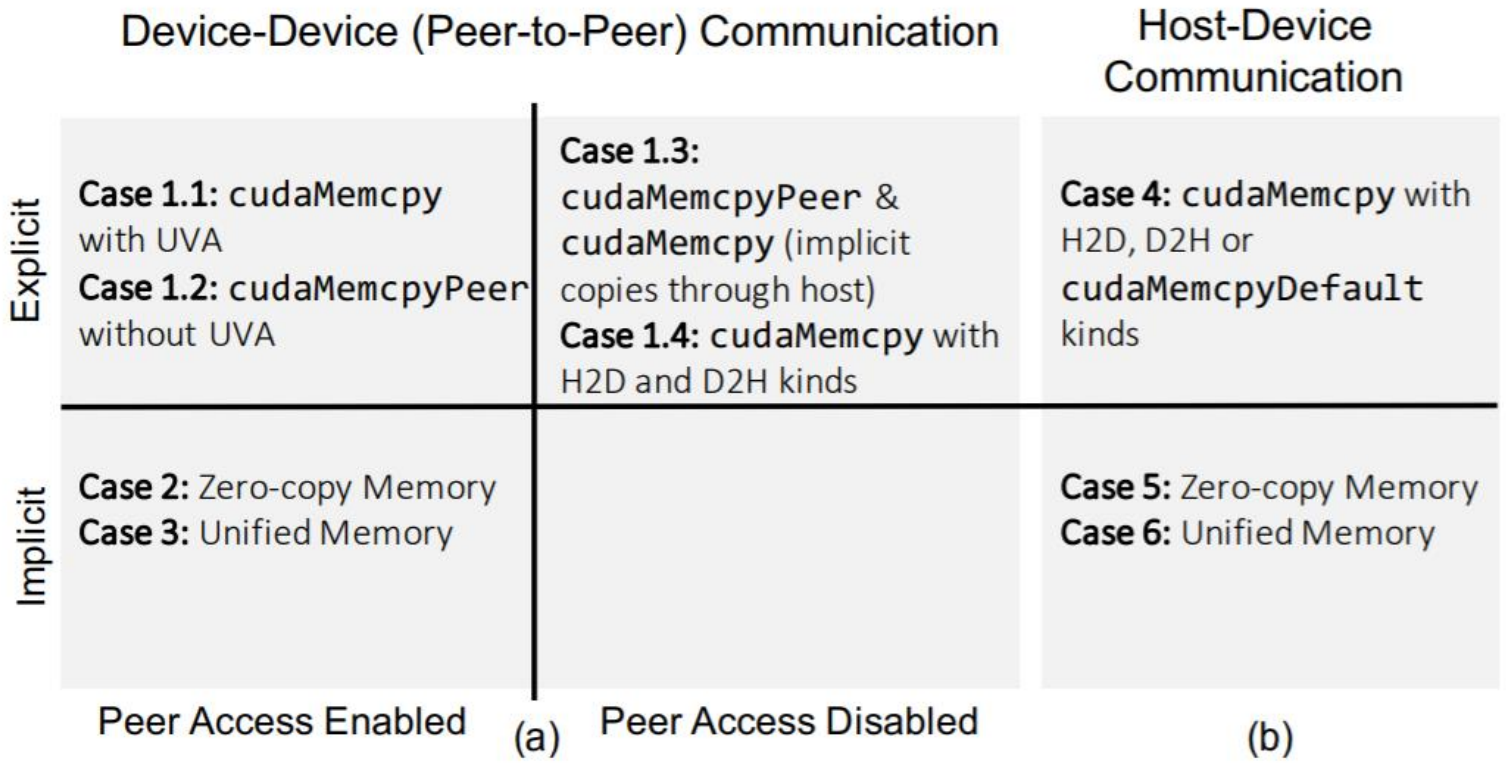
Point-to-Point

Hyper-cube Mesh

All-to-all

Increasing # of Peers

Different Types of Communication



Why need a communication detection tool for GPUs?

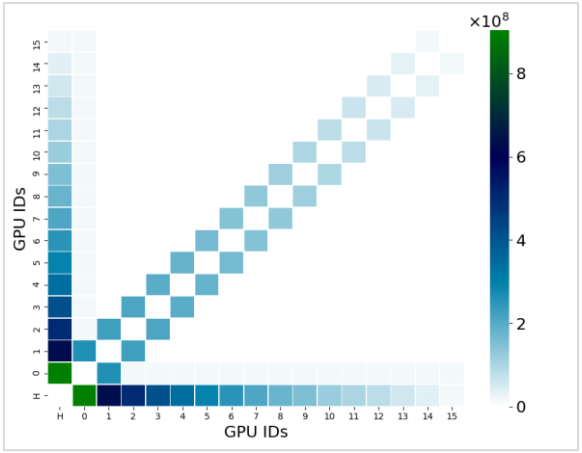
1. Unforeseen communication
   - Implicit data transfers due to different device capabilities and topology
2. Communication bugs
   - Communication bugs in application results in unexpected communication pattern
3. Scalability and performance issues
   - Single GPU applications naively scale on multi-GPUs, can ineffectively use interconnects

Related Work

- **EZTrace**: Generic trace generation framework [1]
- **Comm|Scope**: micro-benchmarks to measure latency and bandwidth for CUDA data transfer primitives [2]
- **Tartan**: multi-GPU benchmark suite for characterizing GPU interconnects [3,4]
- Tools generating communication patterns for multi-core architectures
  - **ComDetective**, leverages PMU and debug registers [5]
  - **Numalize**, uses binary instrumentation [6]

Contributions

- Present **ComScribe**, a tool for generating communication matrices
  - Built on top of NVIDIA's profiling tool *nvprof* [7]
  - Shows both the number of transfers and amount of data transferred between every GPU-GPU and CPU-GPU pair in a node.
  - Identifies different types of communication



- Classification of all types of communication options available for intra-node communication using CUDA
- Evaluation on several micro- and macro-benchmarks as well as two deep learning applications

Our Tool: ComScribe

- Built on top of NVIDIA's profiling tool *nvprof*
- Our tool works in two steps:
  - First, it collects intra-node multi-GPU and CPU-GPU memory transfer information during execution with nvprof.
  - Then, it performs post-processing to quantify communication among GPUs as well as the host, and identify communication types
    - Communication Matrices for Explicit transfers and UM
    - Bar-chart for Zero-copy Memory

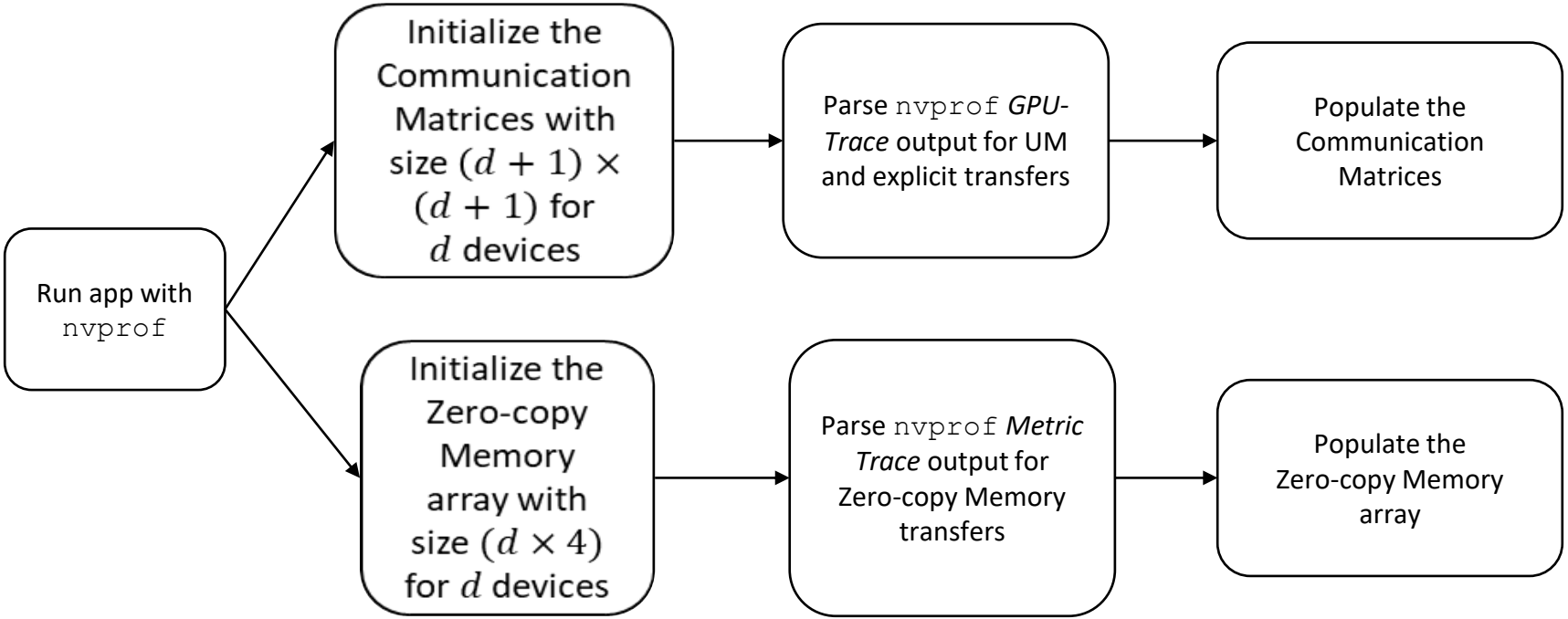
nvprof Pros and Cons

- Pros:
  - Provides a timeline of all activities taking place on a GPU
    - Kernel execution, memory copy, data transfers
  - Has different profiling modes:
    - GPU-Trace Mode - Explicit transfers and UM
    - Metrics Mode - Zero-copy memory transfers

- Cons:
  - Gives trace consisting of **extraneous information**
  - Total amount of data shared between each pair of devices for each type of communication is **not readily available**
  - Requiring **extra effort** by the programmer to extract such information
  - Does **not generate communication matrices**

Our Tool: ComScribe

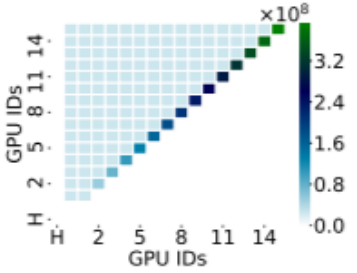
- ComScribe Workflow



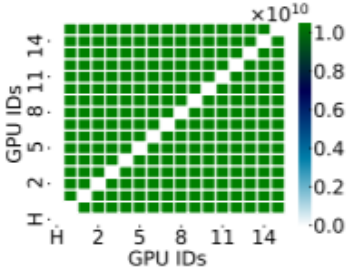
Evaluation

- Validated our tool on
  - 8 micro-benchmarks
    - Comm|Scope [2]
    - MGBench [8]
  - 3 macro-benchmarks
    - NVIDIA's Multi-GPU Jacobi Solver [9]
    - NVIDIA's Monte Carlo Simulation 2D Ising-GPU [10]
    - MGBench's Game of Life [8]
- Insightful Communication Matrices for 2 DNN models
  - Eidetic 3D LSTM (E3D-LSTM) [11]
  - Transformer [12]
- Hardware platform
  - DGX-2 system
    - 16 NVIDIA Tesla V100 GPUs
    - All-to-all topology
- Software Configuration
  - NVIDIA CUDA v10.0.130
  - Python 3.7 and it's packages

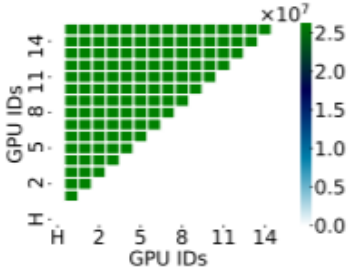
Evaluation (Micro-benchmarks)



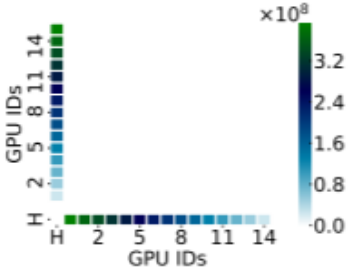
(a) Case 1.1
Full-Duplex (Buggy)



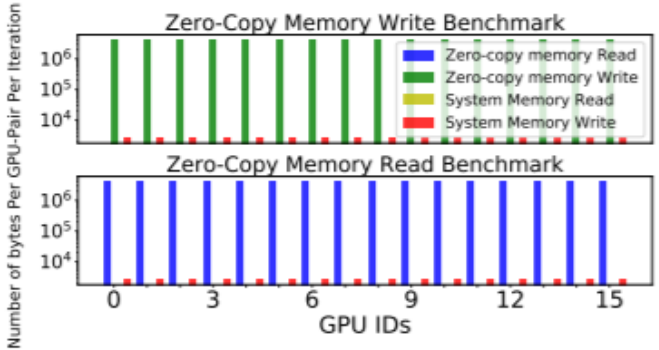
(b) Case 1.2
Full-Duplex



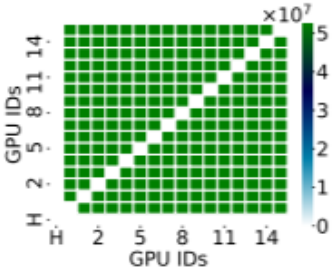
(c) Case 1.2
Half-Duplex



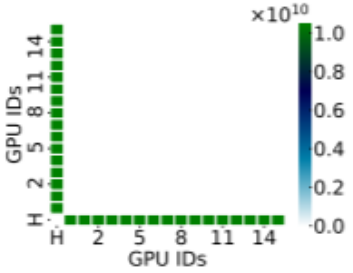
(d) Case 1.3
Half-Duplex



(e) Case 2 and Case 5
Half-Duplex

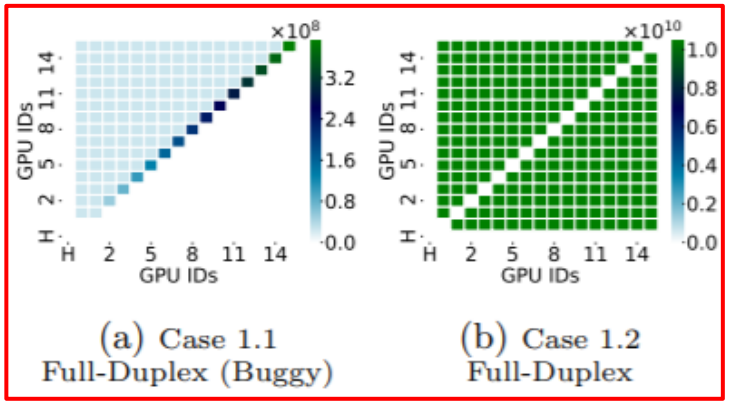


(f) Case 3
Full-Duplex



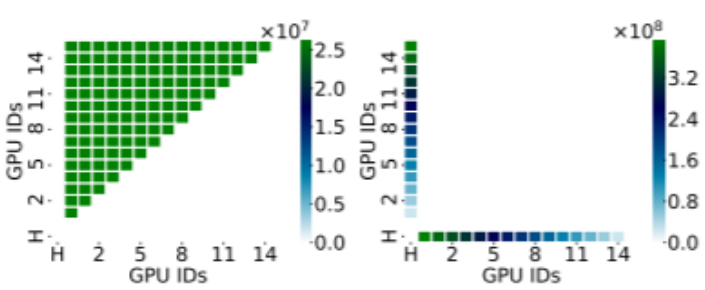
(g) Case 4
Half-Duplex

Evaluation (Micro-benchmarks)



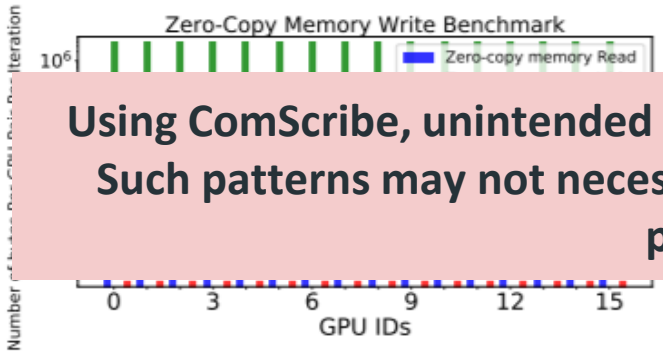
(a) Case 1.1
Full-Duplex (Buggy)

(b) Case 1.2
Full-Duplex

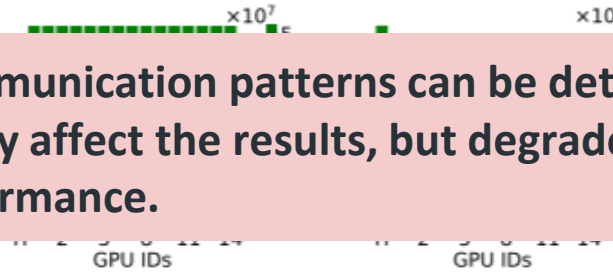


(c) Case 1.2
Half-Duplex

(d) Case 1.3
Half-Duplex



(e) Case 2 and Case 5
Half-Duplex

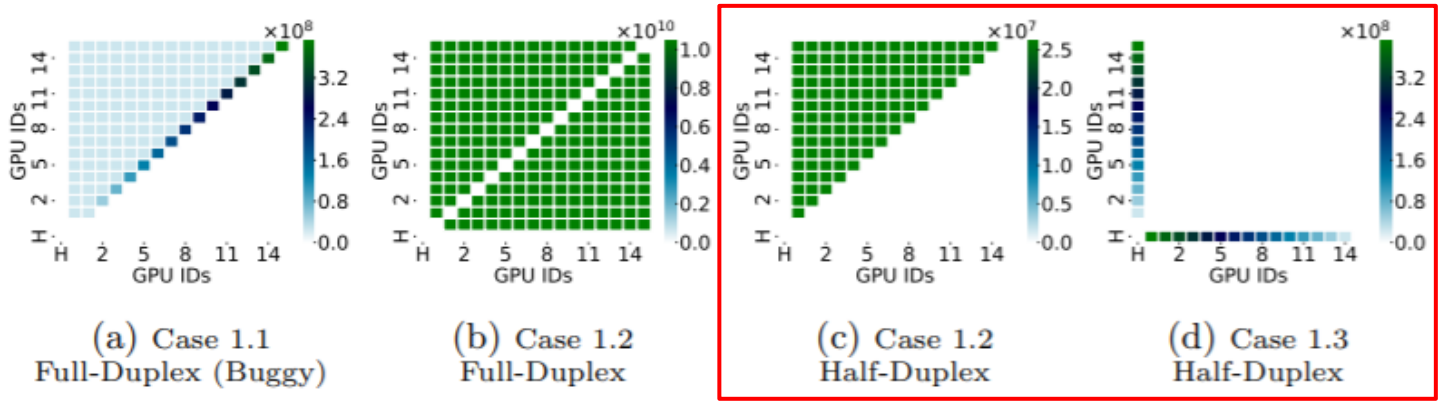


(f) Case 3
Full-Duplex

(g) Case 4
Half-Duplex

Using ComScribe, unintended communication patterns can be detected. Such patterns may not necessarily affect the results, but degrade the performance.

Evaluation (Micro-benchmarks)



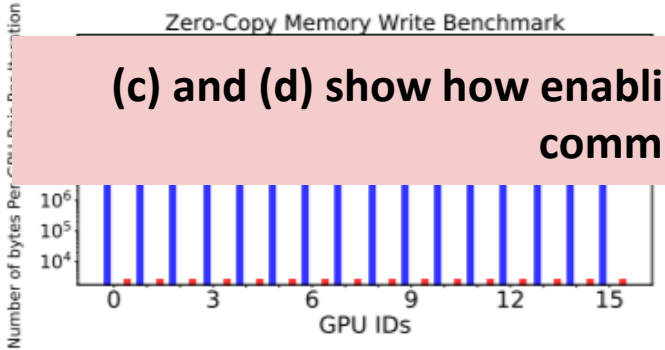
(a) Case 1.1 Full-Duplex (Buggy)

(b) Case 1.2 Full-Duplex

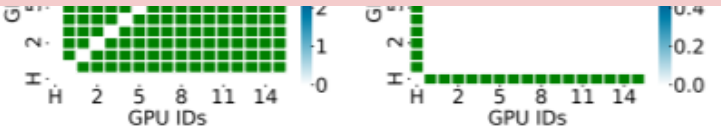
(c) Case 1.2 Half-Duplex

(d) Case 1.3 Half-Duplex

(c) and (d) show how enabling and disabling peer-access affects the communication pattern.



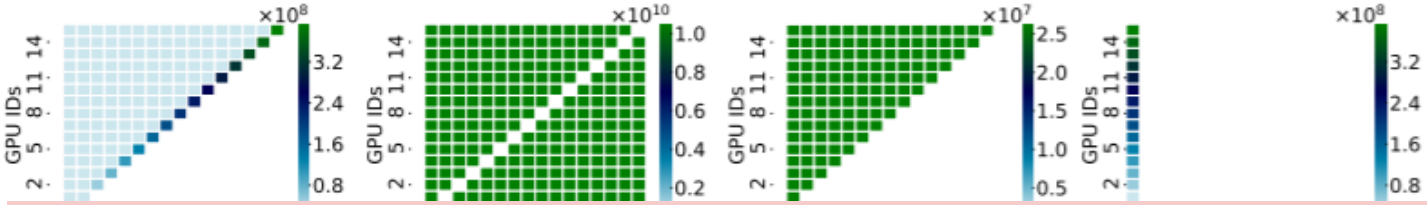
(e) Case 2 and Case 5 Half-Duplex



(f) Case 3 Full-Duplex

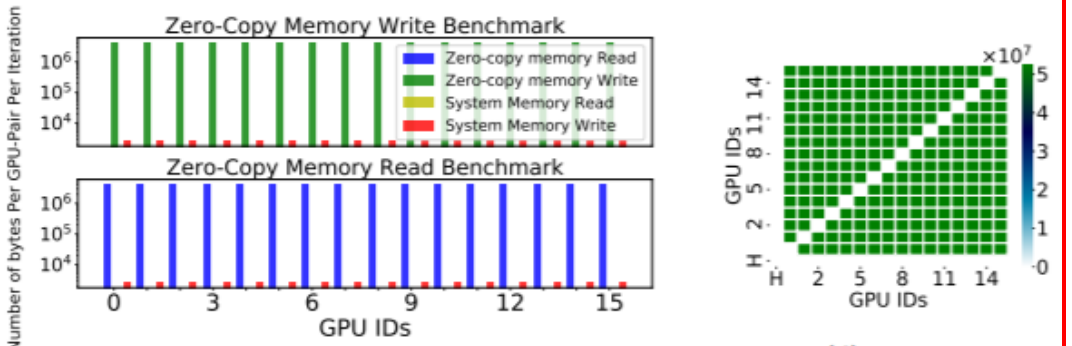
(g) Case 4 Half-Duplex

Evaluation (Micro-benchmarks)



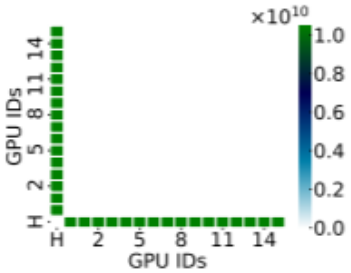
ComScribe can identify implicit transfers as well. Zero-copy memory transfers are presented as a bar-chart.

Full-Duplex (Buggy) Full-Duplex Half-Duplex Half-Duplex



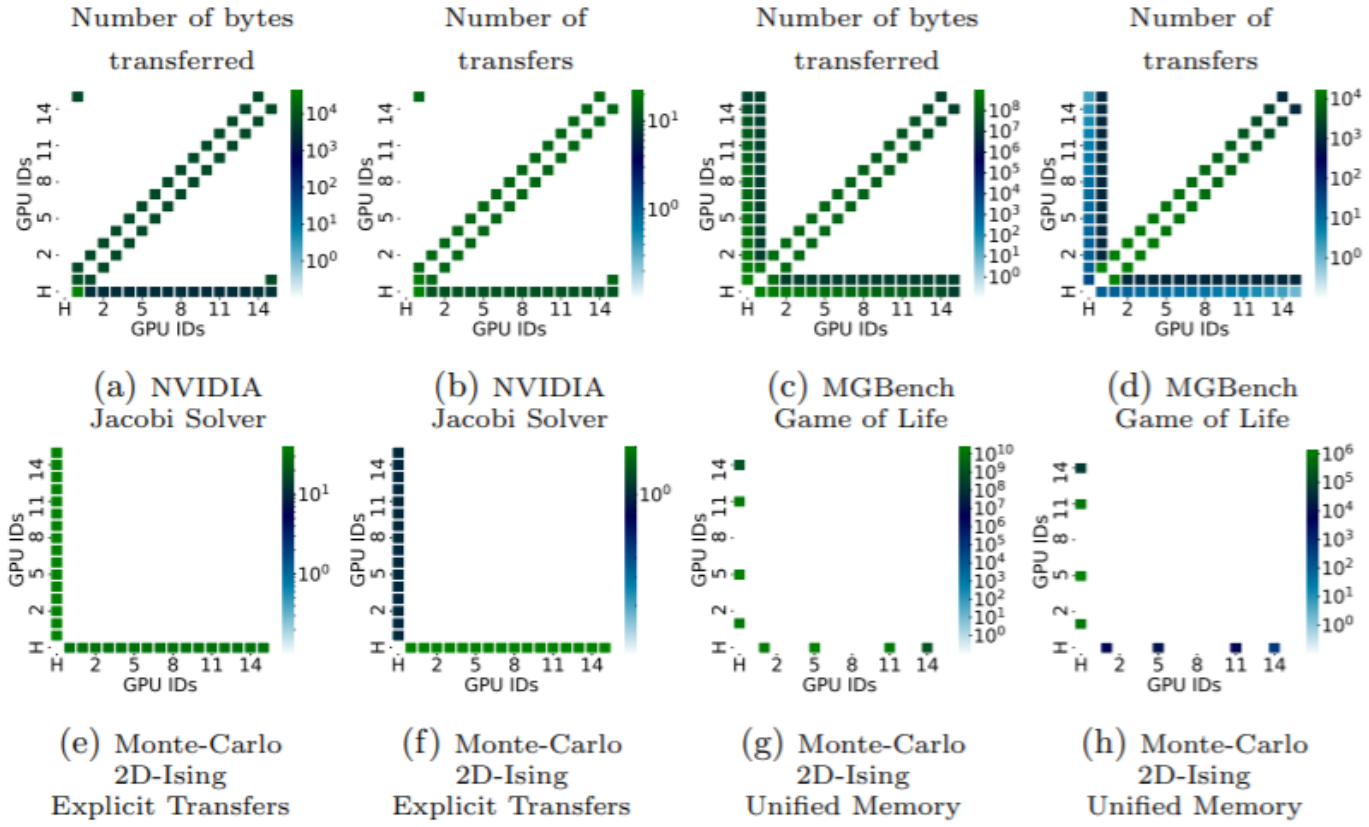
(e) Case 2 and Case 5
Half-Duplex

(f) Case 3
Full-Duplex

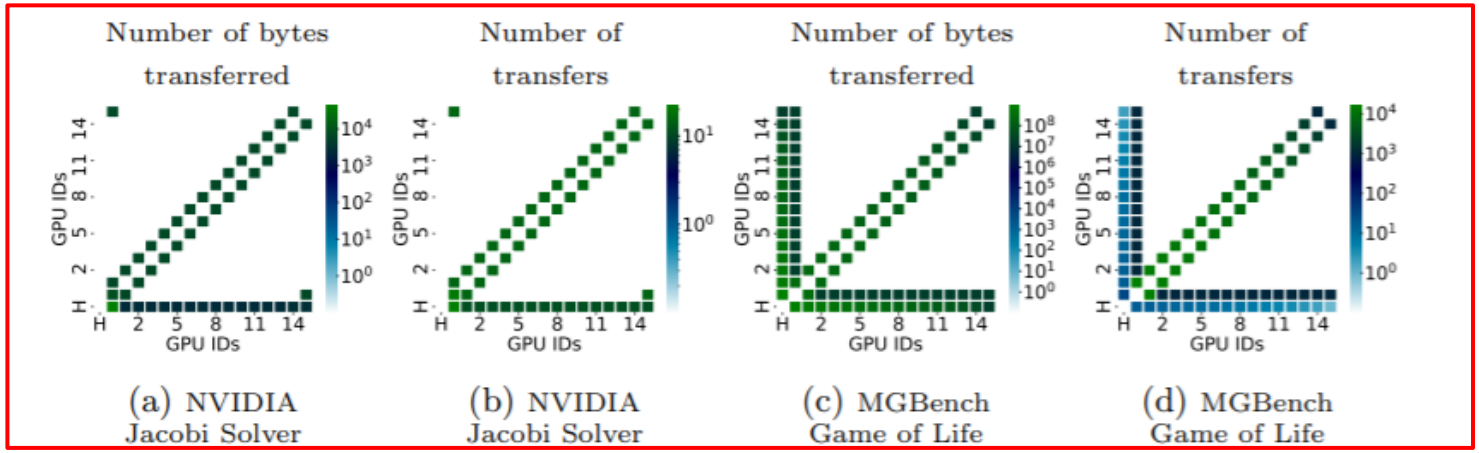


(g) Case 4
Half-Duplex

Evaluation (Macro-benchmarks)



Evaluation (Macro-benchmarks)



Both Jacobi Solver and Game of Life employ a nearest-neighbor communication pattern, which can be easily observed from the communication matrices.

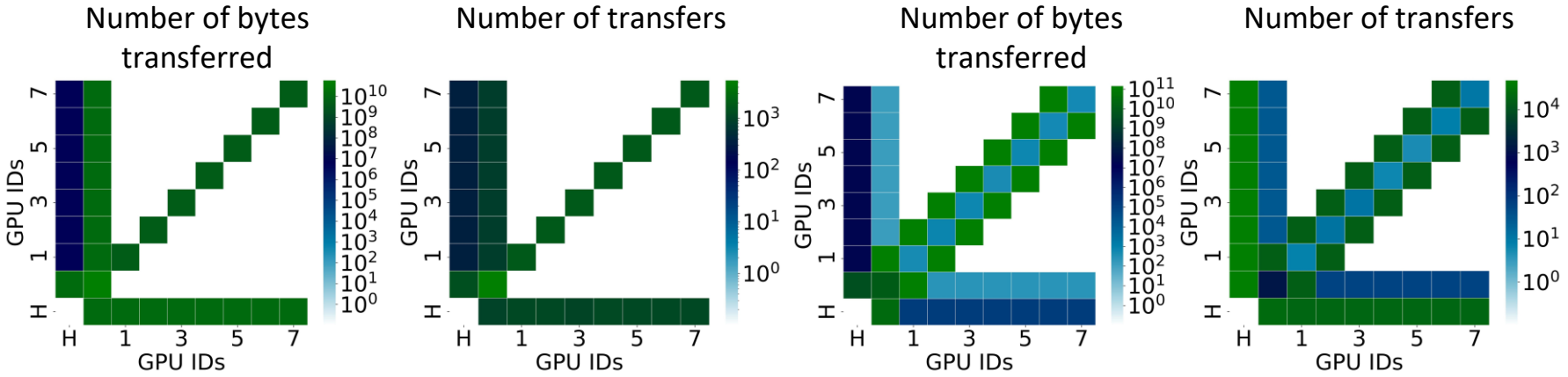
(e) Monte-Carlo 2D-Ising Explicit Transfers

(f) Monte-Carlo 2D-Ising Explicit Transfers

(g) Monte-Carlo 2D-Ising Unified Memory

(h) Monte-Carlo 2D-Ising Unified Memory

Evaluation (DNN models)



a) Eidetic 3D LSTM b) Eidetic 3D LSTM c) Transformer d) Transformer

From the communication matrices of DNN modes (E3D LSTM and Transformer), two variations of communication used for implementing data parallelism can be observed.

Conclusion

- We developed ComScribe on top of NVIDIA's profiling tool nvprof that
  - identifies, quantifies and generates communication matrices for GPU-GPU and CPU-GPU communications in a single node
- Communication matrices generated by our tool can be used by programmers to
  - differentiate types of communication
  - study the communication patterns
  - detect communication bugs in a multi-GPU application

References

[1] Trahay, F., Rue, F., Faverge, M., Ishikawa, Y., Namyst, R., Dongarra, J.: Eztrace:a generic framework for performance analysis. In: 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. pp. 618–619. IEEE (2011)

[2] Pearson, C., Dakkak, A., Hashash, S., Li, C., Chung, I.H., Xiong, J., Hwu, W.M.:Evaluating characteristics of cuda communication primitives on high-bandwidth interconnects. In: Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering. p. 209–218. ICPE '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3297663.3310299

[3] Li, A., Song, S.L., Chen, J., Li, J., Liu, X., Tallent, N.R., Barker, K.J.: Evaluating modern gpu interconnect: Pcie, nvlink, nv-sli, nvswitch and gpudirect. IEEE Transactions on Parallel and Distributed Systems 31(1), 94–110 (2020)

[4] Li, A., Song, S.L., Chen, J., Liu, X., Tallent, N., Barker, K.: Tartan: Evaluating modern gpu interconnect via a multi-gpu benchmark suite. In: 2018 IEEE International Symposium on Workload Characterization (IISWC). pp. 191–202 (2018)

[5] Sasongko, M.A., Chabbi, M., Akhtar, P., Unat, D.: Comdetective: a lightweight communication detection tool for threads. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. pp. 1–21 (2019)

[6] Diener, M., Cruz, E.H., Alves, M.A., Navaux, P.O.: Communication in shared memory: Concepts, definitions, and efficient detection. In: 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP). pp. 151–158. IEEE (2016)

[7] NVIDIA: Cuda profiler user's guide. https://docs.nvidia.com/cuda/pdf/CUDA Profiler Users Guide.pdf (July 2020)

[8] Ben-Nuun, T.: Mgbench: Multi-gpu computing benchmark suite (cuda).https://github.com/tbennun/mgbench (2017)

[9] NVIDIA: Multi-gpu-programming-models: Examples demonstrating available options to program multiple gpus in a single node or a cluster. https://github.com/NVIDIA/multi-gpu-programming-models (2018)

[10] NVIDIA: Ising-gpu: Gpu-accelerated monte carlo simulations of 2d ising model. https://github.com/NVIDIA/ising-gpu (2019)

[11] Wang, Y., Jiang, L., Yang, M.H., Li, L.J., Long, M., Fei-Fei, L.: Eidetic 3d LSTM:A model for video prediction and beyond. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=B1lKS2AqtX

[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser,L., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)

Thank You!

ComScribe is available at
https://github.com/ParCoreLab/ComScribe

Questions?