MAccel-sim: A multi-gpu simulator for architectural exploration

Christin Bose, Cesar Avalos, Junrui Pan, Mahmoud Khairy*, Timothy Rogers
Elmore Family school of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, USA
*Currently at AMD Research

***Abstract*—Graphical processing units (GPUs) have found use in plethora of modern day applications such as machine learning and data analytics. GPU architectural simulators have been built to enable hardware-software codesigning to extract peak performance and performance per watt from such architectures. Many applications such as recommendation models and graph neural networks benefit from the use of multi-GPUs to scale up the size of the workload and/or processing throughput. Current open-sourced GPU architectural simulators have traditionally not been able to efficiently model multi-GPU workloads that can cater to a wide variety of applications. In computer architecture, it is no secret that innovation is often powered by the industry. However, the simulation tools used by industry are often closed sourced and hence limiting in the goal to democratize architectural research. This paper proposes an initial design of MAccel-sim that extends Accel-sim to model a multi-GPU system. We highlight the limitations of popular state-of-the-art GPU architectural simulators and propose a novel trace-based simulation tool that enables end-to-end simulation of workloads running popular machine learning frameworks like Pytorch or Tensorflow. To conclude, we present correlation studies performed using benchmarks running on a 4xV100 GPU system. We aim to open-source this work to democratize architectural studies and further push the envelope of GPU research.**

***Index Terms*—GPU architecture, modelling, Multi-GPU systems, NVbit**

I. Introduction

GPUs have been used to accelerate workloads that are commonly used in deep learning and exascale computing systems. Typically, such workloads exhibit high levels of implicit parallelism and hence are amenable to increased performance scalability as long as GPUs can scale their hardware resources. Emerging types of workloads such as recommendation models and graph neural networks often have memory footprints that are much larger than what a single GPU can provide and hence are often deployed in a multi-GPU system.

To enable hardware-software optimizations for a multi-GPU system and also test out novel architectural ideas, it thus becomes imperative to have a simulator that supports multi-GPU modeling without compromising much on simulation speed. However, while various single GPU architectural simulators have seen continued popularity in the open-sourced community, options for multi-GPU simulators are limited. The prime simulation frameworks to explore multi-GPU architectural ideas in recent top conference venues are [6] and [5]. While simulators from the industry are often flexible and fast (for eg: [6]), they are closed-sourced and hence not available to the broader community. While primarily based on the AMD ISA, MGPUsim [5] requires workloads to be custom re-written which limits the flexibility of such simulators for new workloads. An ideal key feature of a simulator would be to simulate kernels based on representative GPU traces through a popular programming framework such as Pytorch or Tensorflow. GPU microarchitectural bottlenecks on emerging workloads can be identified by being able to simulate the full stack of kernels generated on real traces. Deployment on a multi-GPU system brings with it a whole suite of inter-GPU communication patterns (for eg: through peer-to-peer (P2P) memcopies or NCCL [3]). A recent study [4] reveals that communication will play an increasingly large role (40-75%) in a distributed training setup as model parameters continue to scale. Thus a true multi-GPU simulator must take into account such communication primitives which are commonly used during distributed training of machine learning models.

TABLE I
Comparison of multi-GPU simulators

	NVarchsim [6]	**mgpu-sim [5]**	**MAccel-sim**
Open sourced?	No	Yes	Yes
Workload generation	Unknown	Manual	Trace based
Simulation speed	Fast	Medium	Medium
GPU architecures modelled	NVIDIA	AMD	NVIDIA

II. System overview

We propose MAccelsim, a multi-GPU simulator based on the open-sourced GPU simulator Accelsim [2]. We extend the usage of the NVbit tracer [7] to trace multi-GPU workloads from actual hardware. Our methodology shown in Figure 1 shows how an arbitrary multi-GPU workload can be traced and executed on the first open-sourced multi-GPU simulation framework that can simulate kernels end to end based on popular machine learning frameworks such as Tensorflow or Pytorch. Our framework supports several commonly occurring GPU interconnection topologies such as Ring, Switch-based

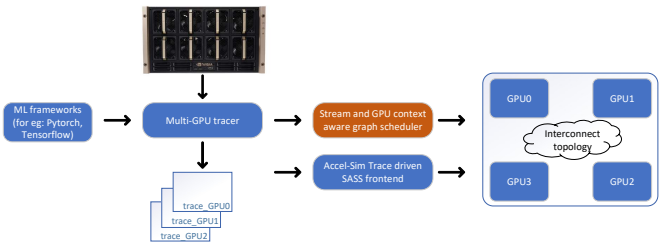

Fig. 1. Proposed multi-GPU simulator.

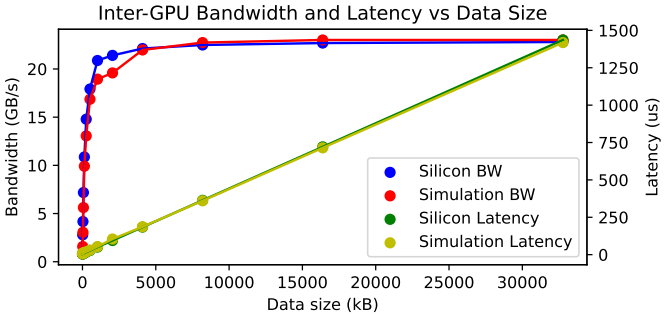

Fig. 2. Inter-GPU bandwidth and Latency test.

and All-to-all topologies through the popular network simulator Booksim [1].

III. Correlation

In this section, we look at examples of using the proposed simulator. The multi-GPU system parameters assumed here are listed in table II. It is imperative that the parameters chosen for modeling closely reflect the hardware specs for the purpose of correlation. The hardware used in our experiments is a 4xV100 NVIDIA GPU system.

TABLE II
Multi-GPU configuration

#GPUs	4
#SMs	80 SMs per GPU
SM configuration	Volta-like SM, 64 warps, 4 warp scheds, 64KB shared memory, 64KB L1 cache, 1.4Ghz
Inter-GPU Interconnect	All-to-all topology, 46 GB/s per link (bi-directional)
Memory BW	1440 GB/s (bidirectional) per GPU

We first validate the inter-GPU latency and bandwidth modeled through a benchmark that performs SM initiated copies through (Unified Virtual Addressing) between GPUs. As shown in figure 2, the inter-GPU bandwidth and latency shows high fidelity with the silicon measurements for various data copy sizes. As expected, at low data copy sizes, the operation is latency-limited and at high data copy sizes, the operation is bandwidth-limited.

The ReduceScatter operation performs a reduction operation on a vector of data and scatters the result in block sized chunks among the various ranks (for eg: GPUs) of a compute

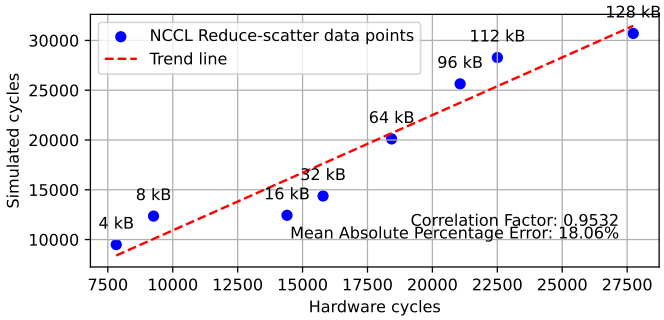

Fig. 3. Validation of Multi-GPU Reduce-scatter NCCL kernel for various data sizes.

system. The ReduceScatter operation is an important phase of the AllReduce kernel as commonly seen in machine learning workloads during gradient synchronization. Figure 3 shows the validation of the ReduceScatter NCCL kernel for various data sizes. The errors in simulation mostly come from the memory accesses to the host which incur a long latency penalty due to traversing the PCIe interconnect (which is not modeled in the proposed multi-GPU simulator). Empirically, we found that the number of such host accesses increases beyond a data size of 128kB giving rise to poor correlation on hardware. Improving the correlation of NCCL kernels remains an area for future work.

IV. Conclusions

Current GPU architectural studies on multi-GPU systems are limited in flexibility outside of the industry. This work proposes a design of a multi-GPU simulator based on Accel-sim [2] that adds much-needed flexibility in the interconnect topology, GPU architectural configuration parameters and can simulate any workload of interest based on actual traces collected from hardware. Future improvements include improving correlation on NCCL kernels and integrating kernel sampling methods to improve simulation speed. We aim to open-source our work once it has undergone rigorous validation.

References

[1] N. Jiang, D. U. Becker, G. Michelogiannakis, J. Balfour, B. Towles, D. E. Shaw, J. Kim, and W. J. Dally, "A detailed and flexible cycle-accurate network-on-chip simulator," in *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2013, pp. 86–96.

[2] M. Khairy, Z. Shen, T. M. Aamodt, and T. G. Rogers, "Accel-sim: An extensible simulation framework for validated gpu modeling," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, 2020, pp. 473–486.

[3] NVIDIA Corp., "NCCL library," https://docs.nvidia.com/deeplearning/nccl.

[4] S. Pati, S. Aga, M. Islam, N. Jayasena, and M. D. Sinclair, "Tale of two cs: Computation vs. communication scaling for future transformers on future hardware," in *2023 IEEE International Symposium on Workload Characterization (IISWC)*, 2023, pp. 140–153.

[5] Y. Sun, T. Baruah, S. A. Mojumder, S. Dong, X. Gong, S. Treadway, Y. Bao, S. Hance, C. McCardwell, V. Zhao, H. Barclay, A. K. Ziabari, Z. Chen, R. Ubal, J. L. Abellán, J. Kim, A. Joshi, and D. Kaeli, "Mgpusim: Enabling multi-gpu performance modeling and optimization," in *Proceedings of the 46th International Symposium on Computer Architecture*, ser. ISCA '19. New York, NY, USA:

Association for Computing Machinery, 2019, p. 197–209. [Online]. Available: https://doi.org/10.1145/3307650.3322230

[6] O. Villa, D. Lustig, Z. Yan, E. Bolotin, Y. Fu, N. Chatterjee, N. Jiang, and D. Nellans, "Need for speed: Experiences building a trustworthy system-level gpu simulator," in *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2021, pp. 868–880.

[7] O. Villa, M. Stephenson, D. Nellans, and S. W. Keckler, "Nvbit: A dynamic binary instrumentation framework for nvidia gpus," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '52. New York, NY, USA: Association for Computing Machinery, 2019, p. 372–383.