Balancing Performance Against Cost and Sustainability in Multi-Chip-Module GPUs

Shiqing Zhang†, Mahmood Naderan-Tahan†, Magnus Jahre‡, Lieven Eeckhout†
†Ghent University, Belgium ‡Norwegian University of Science and Technology (NTNU), Norway

***Abstract*—MCM-GPUs scale performance by integrating multiple chiplets within the same package. How to partition the aggregate compute resources across chiplets poses a fundamental trade-off in performance versus cost and sustainability. We propose the *Performance Per Wafer (PPW)* metric to explore this trade-off and we find that while performance is maximized with few large chiplets, and while cost and environmental footprint is minimized with many small chiplets, the optimum balance is achieved with a moderate number of medium-sized chiplets. The optimum number of chiplets depends on the workload and increases with increased inter-chiplet bandwidth.**

I. Introduction

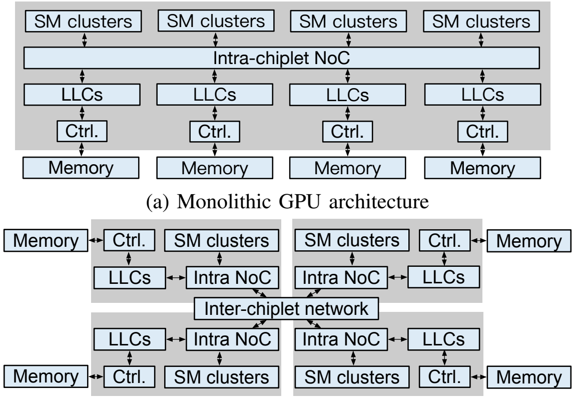
The ever-increasing compute demand of emerging general-purpose GPU applications pushes industry towards GPUs with ever-higher compute capability and thus more Streaming Multiprocessors (SMs). Unfortunately, technology scaling has slowed down significantly, which implies that we can no longer rely on it to scale performance. Increasing die size beyond the reticle limit (around $860\,\text{mm}^2$) is impossible, and furthermore, large die sizes result in lower yield. The result of these trends is the introduction of advanced packaging and stacking solutions to continue to scale performance [11]. One option (explored in this work) is multi-chip-module (MCM) integration in which a GPU package consists of multiple GPU dies, called chiplets, alongside 3D-stacked memory chiplets that are interconnected using, for example, a silicon interposer or organic substrate [1]. Each GPU chiplet is connected to a local memory stack and the chiplets are connected to each other through an inter-chiplet network; and memory, while physically distributed, is logically shared, i.e., all SMs in all chiplets can access the entire memory space. The bandwidth offered by the inter-chiplet network is typically lower than the intra-chiplet Network-on-Chip (NoC) bandwidth. As a result, the effective bandwidth for an SM to access a remote memory partition is lower than when accessing local memory.

While MCM-GPUs provide a pathway to scale GPU performance, they expose a fundamental trade-off between performance and yield (and thus cost and sustainability, as we will explain next). Performance is maximized with few large chiplets because this provides higher effective bandwidth between SMs and (remote) memory partitions, while yield is maximized with many small chiplets. The goal of this paper is to explore this trade-off. We therefore introduce a novel metric, namely *Performance Per Wafer (PPW)*, to quantify the overall performance of MCM-GPUs with a target aggregate number of SMs, consisting of identical chiplets taken from a single wafer. Allocating more SMs to each chiplet increases chiplet size which in turn results in fewer chiplets per wafer and the GPU requiring fewer chiplets to reach the target aggregate SM count. Larger chiplets also reduce yield, meaning that a larger proportion of the chiplets in a wafer will be unusable due to production defects. Creating smaller chiplets conversely requires more chiplets to reach the target aggregate SM count, which improves yield. On the flip side, it also means that a larger proportion of SMs may communicate with remote memory partitions using the (limited) inter-chiplet bandwidth.

An important motivation for looking into the performance versus yield trade-off relates to cost and sustainability. Highest performance is achieved with few large chiplets (or even a single chip if possible). Presumably, this is what industry is pursuing (or has been pursuing) in spite of the high cost (both monetary and environmentally) due to the low yield, as testified by the large GPUs on the market, e.g., Nvidia's Hopper GPU with a $814\,\text{mm}^2$ die.[1] At the other end of the spectrum, cost is minimized with many small chiplets [8]. Likewise, the embodied environmental footprint per chiplet is minimized with a small die size [6]. The reason is that the embodied footprint per wafer is high and continues to increase with technology advancements: the amount of energy needed to produce a wafer, the chemicals and gases emitted during manufacturing, the materials needed (some of which are rare and energy-intensive to extract), as well as the ultra-pure water consumption increases with each generation of technology node [3]. Because datacenter infrastructure is mostly dominated by the embodied footprint [7], in part because of the operational energy consumption being empowered by green energy sources, the embodied footprint serves as a proxy for the overall environmental footprint [5].

In this work, we explore this MCM-GPU trade-off in performance versus cost and sustainability. While few large chiplets maximize performance, many small chiplets minimize cost and environmental footprint. We find the optimum PPW configuration to be in the middle with a moderate number of medium-sized chiplets. The optimum depends on the workload's characteristics and the inter-chiplet network topology and bandwidth. We find that for workloads with significant inter-chiplet traffic (due to data sharing across chiplets) the optimum shifts towards fewer chiplets, while for workloads with limited inter-chiplet communication, the optimum shifts towards more chiplets. We find that the design space needs to

[1] The total cost of a system is a function of engineering cost, manufacturing cost (including packaging), and deployment cost (including provisioning of power and cooling). The lack of publicly available data makes it hard, or impossible, to make a detailed cost analysis. This work hence uses yield as a first-order proxy for manufacturing cost.



(a) Monolithic GPU architecture

(b) 4-chiplet GPU architecture

Fig. 1: Architectural resource allocation example comparing an unrealistic monolithic GPU versus a 4-chiplet GPU. *Distributing architectural resources across more chiplets improves yield but exposes SMs to the lower inter-chiplet bandwidth.*

Parameter	Aggregate	Per Chiplet			
		2-chiplet	4-chiplet	8-chiplet	16-chiplet
Number of SMs	256	128	64	32	16
LLC capacity (MB)	64	32	16	8	4
Memory bw. (TB/s)	8	4	2	1	0.5
On-chip NoC bw. (TB/s)	40	20	10	5	2.5
Baseline inter bw. (TB/s)	7.2	3.6	1.8	0.9	0.45
Estimated area (mm^2)	1,600	800	400	200	100

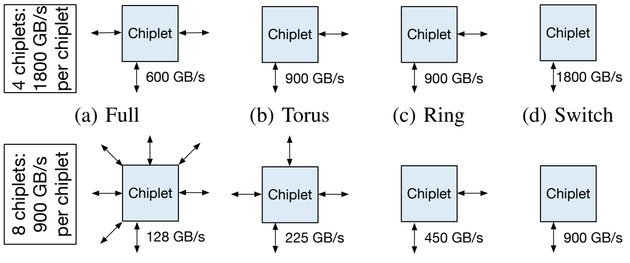
TABLE I: MCM-GPU system configurations in this study.

be explored holistically: perhaps contrary to common intuition, a locally suboptimal component, i.e., a high-bandwidth inter-chiplet network which incurs a higher cost than a low-bandwidth network, leads to a system that globally optimizes performance versus cost and environmental footprint.

II. Problem Statement

Since the area of a wafer is finite, the task at hand is to allocate this limited area to architectural resources across chiplets in a way that maximizes PPW. We now discuss how performance and yield scale with chiplet count.

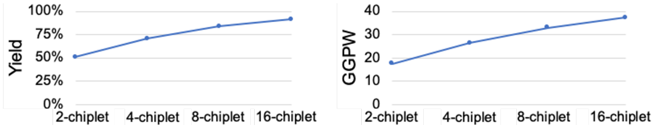
Performance. We consider a target GPU system with a total of 256 SMs, 64 MB Last-Level Cache (LLC) capacity, and 8 TB/s memory bandwidth. Manufacturing this large a monolithic chip, as illustrated in Figure 1a, is unrealistic because its total chip area is estimated to be 1,600 mm^2. Instead, a chiplet-based design enables manufacturing this large a GPU by partitioning the architecture resources across multiple chiplets with each chiplet featuring proportionally fewer architecture resources. We consider 2, 4, 8 and 16 chiplets with proportionally scaled down architecture resources.[2] Figure 1b illustrates the 4-chiplet design with one fourth of the resources per chiplet relative to the monolithic design, i.e., 64 SMs, 16 MB LLC capacity, and 2 TB/s memory bandwidth per 400 mm^2 chiplet. Table I specifies how chiplet resources scale with chiplet count. Note that this is just one way to scale GPU resources through MCM integration. Alternative options include 3D-stacking of cache and compute chiplets, e.g., AMD's V-Cache or 3D-stacking of compute chiplets [11]. Exploring resource scaling alternatives is left for future work.



(a) Full (b) Torus (c) Ring (d) Switch

(e) Full (f) Torus (g) Ring (h) Switch

Fig. 2: Inter-chiplet bandwidth distribution for 4 (top row) and 8 chiplets (bottom row) for different network topologies. *Per-link bandwidth varies with chiplet count and network topology.*



(a) Chiplet yield (b) Good GPUs per wafer

Fig. 3: CY and GGPW as a function of the number of chiplets. *More chiplets implies smaller chiplet size, which leads to higher chiplet yield and more good GPUs per wafer.*

We further assume that the total aggregate bandwidth offered by the silicon interposer or bridges is constant, i.e., yield and cost of the inter-chiplet network is constant [8]. The off-chiplet bandwidth thus decreases proportionally with chiplet count, as reported in Table I ranging from 3.6 TB/s for the 2-chiplet configuration to 450 GB/s for the 16-chiplet configuration. This possibly incurs performance implications for applications with significant inter-chiplet communication.

Finally, we assume a memory-side LLC organization (unless mentioned otherwise), which means that the LLC can only cache data from its local memory partition [1]. A remote memory request needs to traverse the lower-bandwidth inter-chiplet links to access remote LLC/memory of another chiplet. In contrast, an SM-side LLC can cache remote data locally, making it less sensitive to inter-chiplet bandwidth [17].

The available inter-chiplet network bandwidth not only depends on chiplet size (and thus chiplet count), it also depends on the inter-chiplet network topology, as illustrated in Figure 2. Assuming a fully connected network, the available off-chiplet bandwidth is evenly partitioned to all connections going out of the chiplet. Because there are fewer connections going out per chiplet, the per-link bandwidth increases for a torus, ring, and switch network. Note that a switch network needs additional switch chiplets for increased bandwidth [15].

Yield. de Vries [4] provides a formula for the number of *chiplets per wafer (CPW)* as a function of die area A:

$$CPW = \frac{\pi d^2}{4A} - 0.58\frac{\pi d}{\sqrt{A}},$$

where d is the wafer's diameter (e.g., 300 mm^2). We further consider the Murphy yield model [10] to compute *chiplet yield (CY)* as a function of die area A:

$$CY = \left(\frac{1 - e^{-AD_0}}{AD_0}\right)^2,$$

with D_0 the defect density per cm^2, assumed to equal 0.09 for volume production processes for a 5 nm technology node according to TSMC [16]. This enables us to compute the number of *good GPUs per wafer (GGPW)*:

[2] The two-chiplet design features 128 SMs, 32 MB LLC and 4 TB/s for a 800 mm^2 die size per chiplet which is in line with Nvidia's Volta [12], Ampere [13], and Hopper [14] GPUs with 815 mm^2, 826 mm^2, and 814 mm^2 die sizes, respectively.

Benchmark	CTAs	Footprint (MB)	MPKI	Min RPKI	Max RPKI
BFS	1,088	56	93	35	58
B+Tree	3,306	121	4	1	2
LUD	16,128	196	4	0.5	1
DWT2D	30,052	264	7	0.002	0.04

TABLE II: Benchmarks used in this study.

$$GGPW = \frac{CY \times CPW}{N},$$

with N chiplets per GPU. Figure 3 reports CY and GGPW as a function of the number of chiplets per GPU. The key conclusion is that both CY and GGPW increase with an increasing number of chiplets, which, assuming constant aggregate GPU resources, corresponds to increasingly smaller chiplets. More specifically, when increasing the number of chiplets from 2 to 4, 8, and 16, the yield of a single chiplet increases from 51% to 71%, 84%, and 91%, respectively. The number of good GPUs per wafer also increases from 17 to 26, 33, and 37. Real-world designs employ redundancy and bypassing faulty modules to improve the effective yield as well as die speed-binning to maximize profit [8]. We hence use yield as a first-order proxy for manufacturing cost (and embodied footprint).

Putting It Together. With the notion of GGPW, we define *Performance Per Wafer (PPW)* as the product of GGPW and the Instructions Per Cycle (IPC) of each good GPU:

$$PPW = GGPW \times IPC.$$

PPW balances GPU performance against yield/cost/footprint.

III. Experimental setup

We modified Accel-Sim [9] to evaluate MCM-GPU performance. We simulate the configurations as listed in Table I while assuming that each SM features a 228 KB L1 data cache and shared memory; two SMs share a network port to the 40 TB/s on-chip crossbar; sectored cache organization and an HBM3 interface. We further assume first-touch memory page allocation and distributed CTA scheduling [1].

We consider four benchmarks from Rodinia, namely BFS, B+Tree, LUD and DWT2D, with varying degrees of memory intensity, see Table II. MPKI reports the number of L1 cache misses per kilo instructions, while RPKI reports the number of remote misses (i.e., L1 cache misses incurring a remote memory partition access) per kilo instructions; the latter is hence a measure to what extent the workload stresses the inter-chiplet network. RPKI depends on the number of chiplets and hence we report the minimum (2 chiplets) and maximum (16 chiplets). We carefully scale the input sets to provide enough threads and CTAs, making full use of the aggregate resources provided. To limit simulation time, we simulate 1 B instructions for all benchmarks, except BFS (10 M).

IV. Results

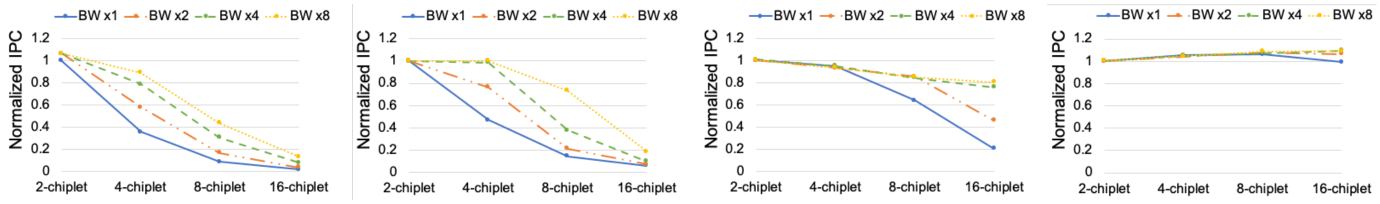
Performance. Figure 4 reports IPC normalized to our baseline crossbar inter-chiplet network with 7.2 TB/s aggregate bandwidth when distributing 256 SMs, 64 MB LLC capacity, and 8 TB/s of memory bandwidth across, 2, 4, 8, and 16 chiplets (see Table I). We also consider GPUs with ×2, ×4, and ×8 our baseline aggregate inter-chiplet bandwidth (14.4, 28.8, and 57.6 TB/s, respectively). The performance trends of each benchmark are explained by their RPKI (Table II) which increases with chiplet count because it becomes increasingly likely that shared data elements are allocated in remote memory partitions. Moreover, the aggregate inter-chiplet bandwidth is distributed across more chiplets which in turn reduces the inter-chiplet bandwidth available to each chiplet proportionally to the number of chiplets. BFS has an RPKI of 35 in the 2-chiplet GPU and 58 in the 16-chiplet architecture, and hence its performance hence degrades significantly with increasing chiplet count (Figure 4a). B+Tree and LUD have (much) lower RPKI which results in smaller slowdowns, especially for the configurations with more inter-chiplet bandwidth. DWT2D has low RPKI across all configurations and its performance is hence insensitive to chiplet count.

Performance Per Wafer. Figures 5 reports PPW across the same design space. Because PPW is the product of IPC and the number of good dies per wafer, the optimum is achieved when the benefit of higher yield is balanced against the reduction in IPC caused by increasingly limited inter-chiplet bandwidth. The optimum chiplet count depends on the workload's characteristics (i.e., degree of inter-chiplet communication due to data sharing) as well as on the available inter-chiplet network bandwidth. Indeed, the 2-chiplet configuration is optimal for the high-RPKI BFS with the ×1 and ×2 bandwidth configurations, whereas the higher performance with the ×4 and ×8 bandwidth configuration shifts the optimum to the 4-chiplet GPU. The performance of the low-RPKI DWT2D on the other hand is insensitive to inter-chiplet bandwidth and PPW hence increases with chiplet count and is maximized for the 16-chiplet configuration. For LUD, the optimum chiplet count shifts from 4 (×1 bandwidth) to 8 (×2) and 16 (×4 and ×8).

Inter-chiplet network topology. We considered a fully connected (Full) inter-chiplet network so far. Figure 6 reports results for Ring, Torus and Switch topologies for the ×1 and ×8 inter-chiplet bandwidth configurations. All results are normalized to the 2-chiplet GPU with ×1 inter-chiplet bandwidth. Higher effective inter-chiplet bandwidth shifts the sweet spot in PPW towards higher chiplet counts. In particular, Switch and Ring achieve optimum PPW with 4 chiplets in the ×1 inter-chiplet configuration (Figure 6a), whereas the 8-chiplet configuration is optimal for these topologies at ×8 bandwidth (Figure 6b). In contrast, the fully connected network is suboptimal because of its lower performance. Switch is suboptimal, especially at ×8 bandwidth because of the additional switch chiplets involved (eight 300 mm^2 chiplets are needed to provide 8 times 7.2 TB/s).

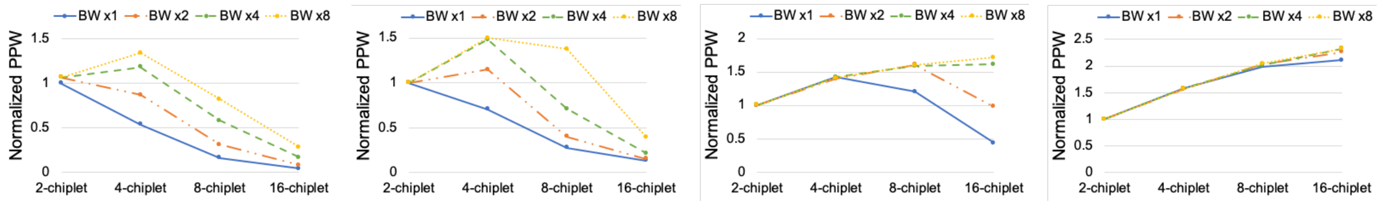
It is further worth noting that, across the entire design space, PPW is maximized at relatively high chiplet count and high inter-chiplet bandwidth, i.e., Torus and Ring at ×8 bandwidth maximizes PPW across the design space, yielding on average 75% higher PPW compared to the baseline 2-chiplet GPU at ×1 bandwidth. This is counter-intuitive perhaps: it shows that a locally suboptimal component, i.e., a high-bandwidth inter-chiplet network which incurs a higher cost than a low-bandwidth network, leads to a globally optimal system, i.e., a high-performance, low-cost/footprint MCM-GPU. Interestingly, Arunkumar et al. [2] reached a similar conclusion when analyzing energy efficiency: they found that a high-energy, high-bandwidth inter-chiplet network is beneficial for reducing an MCM-GPU's total energy consumption.

SM-side LLC. Figure 6 also reports PPW for the Ring



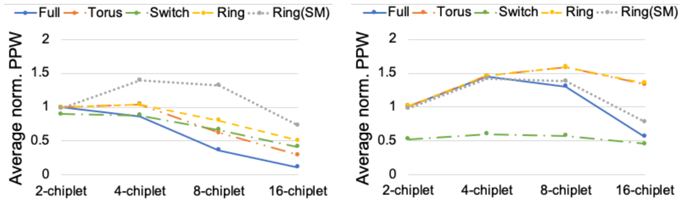
(a) BFS (b) B+Tree (c) LUD (d) DWT2D

Fig. 4: IPC versus chiplet count with a fully connected inter-chiplet network. *The performance trends are determined by RPKI.*



(a) BFS (b) B+Tree (c) LUD (d) DWT2D

Fig. 5: PPW versus chiplet count with a fully connected inter-chiplet network. *Optimal PPW balances yield and performance.*



(a) Bandwidth ×1 (b) Bandwidth ×8

Fig. 6: PPW as a function of chiplet count for (a) the ×1 and (b) ×8 inter-chiplet bandwidth configurations, normalized to the two-chiplet fully connected topology.

topology (the optimum topology) assuming an SM-side LLC. The fundamental trade-off is qualitatively similar for SM-side and memory-side LLCs. However, because the SM-side LLC is less sensitive to inter-chiplet bandwidth, the optimum chiplet count is the same for the ×1 and ×8 configurations, while shifting towards higher chiplet counts under the memory-side LLC. Interestingly, PPW is maximized with an SM-side LLC with 4 chiplets at ×1, while being maximized with a memory-side LLC with 8 chiplets at ×8. The underlying reason is that an SM-side LLC outperforms a memory-side LLC at low inter-chiplet bandwidth, while a memory-side LLC outperforms an SM-side LLC at high inter-chiplet bandwidth.

V. Conclusion

We proposed Performance Per Wafer (PPW) as a novel metric to explore the fundamental trade-off in performance versus cost and sustainability in MCM-GPUs. Performance is maximized with few large chiplets, while cost and environmental footprint is minimized with many small chiplets. The MCM-GPU that optimally balances performance against cost and sustainability features a moderate number of medium-sized chiplets. The optimum number of chiplets depends on the workload and increases with increased inter-chiplet bandwidth.

Acknowledgements

Shiqing Zhang is supported by a CSC scholarship (Grant No. 201903170128), and Magnus Jahre is supported by the Research Council of Norway (Grant No. 286596). Lieven Eeckhout is supported in part by the UGent-BOF-GOA grant No. 01G01421, and the European Research Council (ERC) Advanced Grant agreement No. 741097.

References

[1] A. Arunkumar, E. Bolotin, B. Cho, U. Milic, E. Ebrahimi, O. Villa, A. Jaleel, C.-J. Wu, and D. Nellans, "MCM-GPU: Multi-Chip-Module GPUs for Continued Performance Scalability," in *ISCA*, 2017.

[2] A. Arunkumar, E. Bolotin, D. Nellans, and C.-J. Wu, "Understanding the Future of Energy Efficiency in Multi-Module GPUs," in *HPCA*. IEEE, 2019, pp. 519–532.

[3] M. G. Bardon, P. Wuytens, L.-Å. Ragnarsson, G. Mirabelli, D. Jang, G. Willems, A. Mallik, A. Spessot, J. Ryckaert, and B. Parvais, "DTCO Including Sustainability: Power-Performance-Area-Cost-Environmental Score (PPACE) Analysis for Logic Technologies," in *Proceedings of the International Electron Devices Meeting (IEDM)*, 2020.

[4] D. K. de Vries, "Investigation of Gross Die Per Wafer Formulas," *IEEE Transactions on Semiconductor Manufacturing*, vol. 18, no. 1, pp. 136–139, 2005.

[5] L. Eeckhout, "A First-Order Model to Assess Computer Architecture Sustainability," *IEEE Computer Architecture Letters*, vol. 21, no. 2, pp. 137–140, 2022.

[6] ——, "Kaya for Computer Architects: Towards Sustainable Computer Systems," *IEEE Micro*, vol. 43, no. 1, pp. 9–18, 2023.

[7] U. Gupta, Y. G. Kim, S. Lee, J. Tse, H.-H. S. Lee, G.-Y. Wei, D. Brooks, and C.-J. Wu, "Chasing Carbon: The Elusive Environmental Footprint of Computing," in *HPCA*, 2021.

[8] A. Kannan, N. E. Jerger, and G. H. Loh, "Enabling Interposer-Based Disintegration of Multi-Core Processors," in *MICRO*, 2015, pp. 546–558.

[9] M. Khairy, Z. Shen, T. M. Aamodt, and T. G. Rogers, "Accel-Sim: An Extensible Simulation Framework for Validated GPU Modeling," in *ISCA*, 2020.

[10] R. C. Leachman, "Yield Modeling and Analysis," https://fog.misty.com/perry/cod/references/yield_models.pdf, 2014, [Online; accessed 2023-05-25].

[11] G. H. Loh and R. Swaminathan, "The Next Era for Chiplet Innovation," in *DATE*, 2023, pp. 1–6.

[12] Nvidia, "NVIDIA Tesla V100 GPU Architecture," https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf, 2018, [Online; accessed 2023-05-25].

[13] ——, "NVIDIA A100 Tensor Core GPU Architecture," https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf, 2020, [Online; accessed 2023-05-25].

[14] ——, "NVIDIA H100 Tensor Core GPU Architecture," https://resources.nvidia.com/en-us-tensor-core/gtc22-whitepaper-hopper, 2022, [Online; accessed 2023-05-25].

[15] Nvidia, "NVLINK and NVSwitch," https://www.nvidia.com/en-us/data-center/nvlink/, 2022, [Online; accessed 2023-05-25].

[16] D. Schor, "TSMC 5-Nanometer Update," https://fuse.wikichip.org/news/2879/tsmc-5-nanometer-update/, 2019, [Online; accessed 2023-05-25].

[17] S. Zhang, M. Naderan-Tahan, M. Jahre, and L. Eeckhout, "SAC: Sharing-Aware Caching in Multi-Chip GPUs," in *ISCA*, 2023.