Good things come in small packages: Should we build AI clusters with Lite-GPUs?

Burcu Canakci*, Junyi Liu*, Xingbo Wu*, Nathanaël Cheriere, Paolo Costa,
Sergey Legtchenko, Dushyanth Narayanan, Antony Rowstron
Microsoft Research

ABSTRACT

To match the blooming demand of generative AI workloads, GPU designers have so far been trying to pack more and more compute and memory into single complex and expensive packages. However, there is growing uncertainty about the scalability of individual GPUs and thus AI clusters, as state-of-the-art GPUs are already displaying packaging, yield, and cooling limitations. We propose to rethink the design and scaling of AI clusters through efficiently-connected large clusters of *Lite-GPU*s, GPUs with single, small dies and a fraction of the capabilities of larger GPUs. We think recent advances in *co-packaged optics* can enable distributing AI workloads onto many Lite-GPUs through high bandwidth and efficient communication. In this paper, we present the key benefits of Lite-GPUs on manufacturing cost, blast radius, yield, and power efficiency; and discuss systems opportunities and challenges around resource, workload, memory, and network management.

ACM Reference Format:
Burcu Canakci*, Junyi Liu*, Xingbo Wu*, Nathanaël Cheriere, Paolo Costa, Sergey Legtchenko, Dushyanth Narayanan, Antony Rowstron. 2025. Good things come in small packages: Should we build AI clusters with Lite-GPUs?. In *Workshop on Hot Topics in Operating Systems (HOTOS '25), May 14–16, 2025, Banff, AB, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3713082.3730390

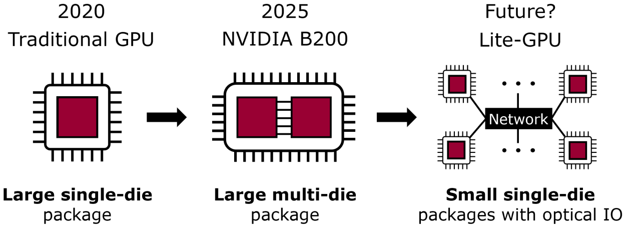


Figure 1: Evolution of GPUs in AI clusters.

1 INTRODUCTION

Demand for AI is growing and expensive to support [34]. These challenges are expected to only get harder as the diversity, complexity, and scale of AI models are growing, making it crucial for AI service providers to build powerful and efficient AI infrastructure [2]. However, scaling AI infrastructure is encountering significant obstacles [37]. We have already reached the limit on how big a compute die can get, leading GPU designers to focus on advanced packaging technologies to pack more transistors into the same package (Figure 1). Nevertheless, scaling an individual GPU package is becoming less and less sustainable for manufacturing due to multiple reasons, including power [55], cooling [21], yield [19, 53], packaging costs [51], and failure blast radius [26]. For instance, the latest generation of NVIDIA GPUs is facing deployment delays due to packaging and cooling issues [18, 52].

We observe that there is an exciting alternative approach to scaling AI clusters. *What if we replace large, powerful GPU packages with highly-connected clusters of Lite-GPUs, that each have only a single, smaller compute die and fractional performance?* Smaller GPUs present many promising hardware characteristics: they have much lower cost for fabrication and packaging, higher bandwidth to compute ratios, lower power density, and lighter cooling requirements. In addition, they can also unlock desirable systems opportunities such as improved fault-tolerance and finer-grained, flexible resource allocation.

To date, distributing AI workloads to large number of GPUs has been challenging due to the data flow demanding very high-bandwidth communication across GPUs [61]. Nevertheless, driven by recent advances in *co-packaged optics*, in the next decade, we expect off-package communication bandwidth to improve by 1–2 orders of magnitude with much better reach (10s of meters), compared to copper-based communication [35, 50, 62]. Co-packaged optics integrates electronic and optical components within millimeters, compared to current pluggable optics, cutting signalling distance and yielding better power efficiency. While there are open

*equal contribution, {burcucanakci,junyili,xingbowu}@microsoft.com.

HOTOS '25, May 14–16, 2025, Banff, AB, Canada

ACM ISBN 979-8-4007-1475-7/2025/05
https://doi.org/10.1145/3713082.3730390

questions and active research on utilizing co-packaged optics, we think that it has the potential to disrupt the trade-off space around designing AI infrastructure. Notably, in the most recent GPU Technology Conference (GTC), NVIDIA highlighted their advancements in co-packaged optics in their effort to massively scale AI infrastructure with much improved power efficiency [38]. We think co-packaged optics can enable Lite-GPUs that are equipped with high-bandwidth and energy-efficient optical interconnects, to communicate with many far-off Lite-GPUs at petabit per second bandwidths [35, 50].

In this paper, we look at AI infrastructure through the lens of Lite-GPUs. Though we give an overview of recent hardware trends and of the key hardware benefits of Lite-GPUs, we focus mainly on the systems opportunities and challenges that would arise as we include Lite-GPUs in AI infrastructure. We discuss how Lite-GPUs could be beneficial in improving resource customization, resource utilization, power management, performance efficiency, and failure blast radii in an AI cluster. In addition, as an initial assessment, we do a performance analysis of a Lite-GPU cluster using popular large language model (LLM) inference workloads. We show that Lite-GPUs have the potential to match or achieve better performance compared to existing GPUs, as they exploit hardware potentials offered by increased total shoreline bandwidth per compute and reduced power density. These benefits can not be realized for free: we identify key research problems around building a cheap and efficient network, co-designing the AI software stack, and data-center management.

2 THE LITE-GPU

In recent years, state-of-the-art data-center GPUs have been increasing compute FLOPS, memory bandwidth, and network bandwidth to support growing AI workloads. As we have already reached the limit of what can be done with a single die [28], improvements have relied on advanced packaging efforts to pack more transistors into the same GPU. For example, most recently, NVIDIA has featured a multi-die GPU design, using high-bandwidth die-to-die interfaces to bind two dies in its Blackwell GPU platform [55]. Alternatively, AMD has proposed chiplets, breaking up monolithic silicon into smaller specialized chips, co-packaged together through 3D stacking [29]. While these techniques have succeeded in improving GPU performance for their generation, there is not a clear path to scaling them further, and in fact, such complex GPU designs are already leading to several difficulties such as maintaining high yield rates, managing high power consumption, and applying efficient cooling [19, 21, 51, 53]. Additionally, as the die gets larger, its area increases faster than its perimeter ("shoreline") that determines the bandwidth it can utilize. This leads to GPUs with high compute-to-bandwidth ratios, which is not always the best fit for AI workloads and results in compute underutilization [4].

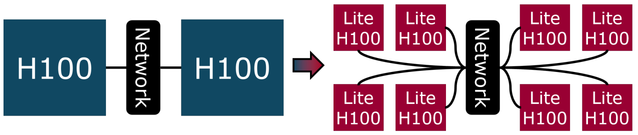


Figure 2: An example Lite-GPU deployment. Each NVIDIA H100 GPU is replaced with four Lite-GPUs, featuring better hardware yield and higher bandwidth-to-compute.

Through Lite-GPUs, we propose an alternative way of scaling AI clusters: with smaller but more GPUs connected through a performant and scalable network, realized through co-packaged optics. A Lite-GPU features a single compute-die GPU package where the die area is much smaller than that of state-of-the-art, leading to several hardware benefits. Figure 2 gives an example of a Lite-GPU system where each NVIDIA H100 GPU is replaced with four Lite-GPUs. In this paper, we mainly use this example while discussing potential benefits of Lite-GPUs in AI clusters.

First, as the die area is smaller per GPU, Lite-GPUs have largely reduced cost of manufacturing due to higher hardware yield rates. For example, the yield rate can be increased by 1.8× when a H100-like compute die area is reduced by $1/4^{th}$, corresponding to almost 50% reduction in manufacturing cost [36].

Second, reducing the compute die area increases the shoreline to die ratio. For example, reducing the die area to $1/4^{th}$ doubles the perimeter exposed to the four dies, yielding a cluster with 2× the bandwidth-to-compute ratio. Although a fraction of the extra bandwidth may be required for additional networking, we show later in our case-study that Lite-GPUs can achieve higher performance efficiency for I/O-bound workloads, such as parts of LLM inference.

Third, smaller packages also greatly reduce complexity of cooling. Today's cutting-edge GPUs already throttle compute frequency to avoid overheating [12, 20]. Smaller single-die GPUs can be air-cooled separately and even sustain higher clock frequencies without requiring advanced cooling.

Overall, we expect the cost of Lite-GPUs to be substantially lower due to better hardware yield and lower packaging costs. While the cost of networking should increase, we expect the net gains to be positive as the networking costs are only a small fraction compared to the GPU costs today. Additionally, there are many active efforts to scale networking costs sublinearly with network size using circuit switching [6, 24], which would allow for even larger Lite-GPU clusters.

3 SYSTEMS OPPORTUNITIES

Consider a cluster of NVIDIA H100 GPUs, which is the most frequently deployed GPU in AI clusters today. Each H100 GPU can be replaced with a number of Lite-H100 GPUs, each Lite-GPU having a fraction of its compute and memory capabilities. Depending on how the Lite-GPUs are customized, compared to the original cluster, the cluster with Lite-GPUs can feature equivalent or better compute, memory, and cost characteristics.

While scaling out AI clusters using today's GPUs with co-packaged optics is an option, as highlighted in the previous section, Lite-GPUs offer many hardware benefits over current GPUs. So in this paper, we focus on utilizing Lite-GPUs as they have potential to unlock the path towards more efficient and scalable AI clusters. Nevertheless, some key systems research questions should be addressed so that we can realize the Lite-GPU disruption.

Scale of distribution Some of the research questions around using Lite-GPUs are not new or unique, but potentially amplified. For example, Lite-GPUs would result in more distributed systems in the datacenter, e.g., small models previously served by a single GPU are now distributed over multiple Lite-GPUs. For larger models that already require multiple GPUs, the number of devices would be multiplied. These can potentially amplify issues such as synchronization and straggling GPUs.

AI clusters come at different scales for training and inference, with training clusters being orders-of-magnitude larger, e.g., 16,000 vs 8 GPUs for Llama 3.1 405B [16, 31]. An inference cluster with Lite-GPUs at the reduction ratios we discussed in Section 2 is unlikely to have more components than a training cluster today and would be easier to realize without heavy innovation in distributing models. In general, building efficient distributed ML training and inference platforms is an active research area and such approaches would also benefit clusters with Lite-GPUs [5, 14, 25, 44].

Finer-granularity of resource management With Lite-GPUs, we can allocate and access smaller units of compute and memory, leading to greater flexibility in managing an AI cluster.

For example, consider *power management*. A GPU's compute clock frequency can be dynamically tuned to lower power consumption during idle periods or to match stragglers [9, 42]. However, the granularity of down-clocking is on all *Streaming Multiprocessors* (SMs). SMs are processors designed for efficient parallel processing and each GPU consists of multiple SMs, similar to cores in a CPU. Down-clocking all SMs of a large GPU can lead to wasted resources or sub-optimal performance. In a Lite-GPU cluster, we can control down-clocking at finer granularity to achieve better power efficiency, akin to down-clocking only a portion of SMs in a larger GPU.

Conversely, we can over-clock Lite-GPUs to achieve higher performance while serving peak workloads, since smaller die areas allow for easier cooling and higher clock frequencies. Alternatively, more Lite-GPUs can be utilized to satisfy the peak load, but with the additional power overhead due to increased networking. Detailed analysis on workload patterns and power modelling can help us determine the most power-efficient approach for serving typical and peak workloads with Lite-GPUs.

Another example of resource management is around *GPU configuration*. Note that today, AI clusters with heterogeneous GPUs are already used to serve requests as power-efficiently as possible, e.g., by deploying different phases of transformer inference on different GPU hardware [40]. We can customize and deploy Lite-GPUs for different profiles of AI workloads, similar to Splitwise, but at much finer scale, e.g., racks of custom Lite-GPUs as opposed to clusters of custom racks. Also, Lite-GPUs can allow for *both* easier over-clocking and higher bandwidth-to-compute ratios, potentially achieving higher performance efficiency at cluster-scale [41, 47].

Third, these smaller GPU units may assist future *AI as a service* offerings. The ability to allocate small customizable Lite-GPU clusters per customer, that are separated physically and provide isolation and security, can be quite powerful.

Workload management Careful workload parallelisation, deployment, and scheduling is a must in order to obtain the benefits of Lite-GPUs and to mask their overheads.

Most importantly, with Lite-GPUs, we move previously in-silicon traffic to an optical network, potentially inducing additional *latency* and *network load*. There are workloads that would be challenging to distribute further using Lite-GPUs, such as workloads that introduce randomness and congestion to the network traffic. Nevertheless, with AI workloads, there are several techniques we can use.

First, AI workloads are highly predictable and pipelined so extra latency can be masked through pre-fetching [15]. In fact, since Lite-GPUs can feature a higher memory bandwidth-to-compute ratio, they may even allow for reduced request-level latency in AI workloads, as less *batching* may be required to improve compute utilization.

Second, large ML models today are already distributed over many GPUs and communicate through highly efficient collectives to minimize the amount of data exchanged, e.g., through tensor parallelism while calculating matrix-matrix multiplications. One can increase the level of tensor parallelism on a deployment of Lite-GPUs to minimize the end-to-end latency.

Fault-tolerance Reducing the size of the GPU naturally reduces the blast radius should a GPU fail due to excessive temperatures, dust or debris, or transistor faults; leading to higher available FLOPS, memory capacity, and memory bandwidth at any time.

To maximize the benefit from smaller blast radii, building a robust and efficient software stack is crucial. Note that today's large-scale inference pipelines already impose larger blast radii than the hardware-imposed blast radii: if one GPU out of a group of GPUs serving a model instance fails, the entire instance is taken offline [24]. Active work on resolving this issue can also help with Lite-GPU clusters [33, 48]. One approach to dealing with such rigid, software-imposed GPU configurations is to include *hot spares*, spare GPUs that can be activated to serve a model instance while recovering from a failure. Lite-GPUs can suit this approach particularly well as a cluster of Lite-GPUs are larger with each additional Lite-GPU being smaller and cheaper. This reduces the proportional overhead of including spare Lite-GPUs, though we still need a strategy for how to best utilize them during normal operation.

In general, Lite-GPUs can help improve fault-tolerance of AI infrastructure. Nevertheless, with Lite-GPUs, the number of GPUs in the cluster is increased and additional networking components may be necessary, potentially leading to different failure frequencies and profiles. A thorough analysis of failures and recovery schemes is necessary to ensure that the reduced blast radius of Lite-GPUs are utilized.

Memory management Each Lite-GPU has only the fraction of the memory capacity of a larger GPU. This can be a problem for workloads that require high memory capacity and do not distribute efficiently. So, there are many open questions about the design of the memory system in a cluster of Lite-GPUs. For instance, do we need memory-sharing across multiple Lite-GPUs to be an option? What should shared memory semantics look like, e.g., do we need to operate with a load/store GPU-to-memory network across Lite-GPUs to prevent extra HBM usage due to network buffering? Additionally, in a heavily-accessed shared memory setting, how can we alleviate the programming and performance challenges that stem from different tiers of memory?

Another potential approach is to use Lite-GPUs along with disaggregated memory [30]. Disaggregated memory can be used to provide a larger memory pool for Lite-GPUs and to allow for more efficient memory sharing across Lite-GPUs, though it introduces additional complexity in memory management. Note that, combined with the finer-granularity of Lite-GPUs, an AI cluster with Lite-GPUs, co-packaged optics, and disaggregated memory can enable us to flexibly adjust the compute-to-memory and compute-to-network ratios per Lite-GPU in the cluster.

Network management Through Lite-GPUs, communication previously in-silicon in a large GPU is now on the Lite-GPU to Lite-GPU network.

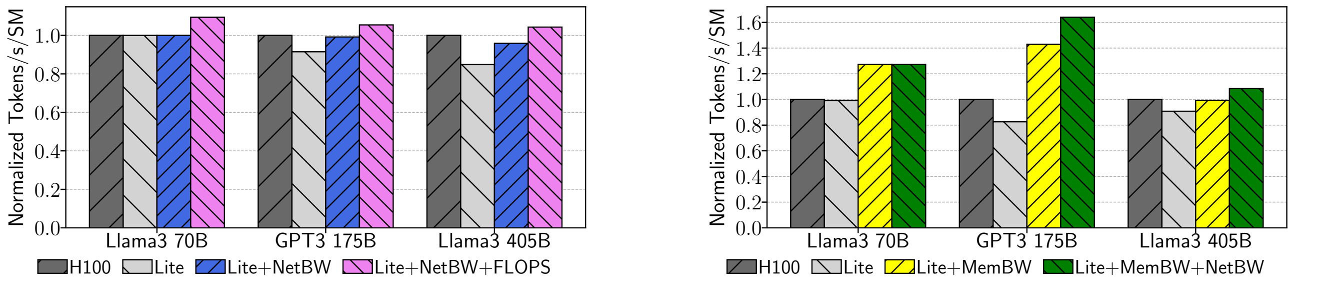
Firstly, the total traffic in a cluster and the total power consumption of the network can be higher. Secondly, in-silicon traffic assumes very high-bandwidth, low latency, and energy-efficient communication. Since the performance and efficiency of communication is degraded outside of silicon, the parallelization and distribution of the workload must be co-designed to minimize the impact of this degradation. Two load/efficiency masking examples (using collectives and prefetching) are mentioned above. Third, with Lite-GPUs, the bandwidth and distance required from GPU-to-GPU links can be higher. Nevertheless, with optical links, we are looking towards petabit per second efficient communication across many racks, which is promising.

With regards to building an efficient, high-bandwidth Lite-GPU network, we have several options. First, as the traffic across Lite-GPUs that replace one large GPU is predictable, we can build a direct-connect topology within that group of Lite-GPUs and leave the remaining network as is. This is an approximation to the original network, though it eliminates the benefits of the smaller blast radius of Lite-GPUs. Alternatively, we can consider a (flat or hierarchical) switched network for the entire Lite-GPU cluster, yielding flexibility and improved fault-tolerance. Using circuit switching, in part or cluster-wide, may be crucial to achieve such a network at low cost. Circuit switching presents the following benefits over packet switching: (i) more than 50% better energy efficiency, (ii) lower latency, and (iii) more ports at high bandwidth, which allows for larger and flatter networks [6].

Data-center management With Lite-GPUs, the number of devices per area is increased, however, the energy per unit area is decreased. There is active research to handle data-center management at scale using various automation techniques which can be applicable to Lite-GPU clusters [22]. Additionally, though the number of devices per rack may increase, the overall cooling requirements of the rack can be lighter due to the more efficient cooling of Lite-GPUs combined with co-packaged optics. This can eliminate the need for liquid cooling racks in the data-center, which comprise a significant portion of racks, and thus space, in an NVIDIA B200 cluster [1].

4 CASE STUDY: LLM INFERENCE

In this section, we present a case study of Lite-GPUs in the context of a trending AI workload — LLM inference [56]. LLM inference involves two distinct phases. The prompt prefill phase processes input tokens to compute reusable intermediate states, i.e., the Key-Value (KV) cache, and generates



(a) Prompt prefill. All configurations perform similarly. As the model sizes grow, the "Lite" cluster underperforms due to increased collectives causing network bottlenecks. Increasing the network bandwidth compensates the increased network demand, overclocking improves performance further as prefill workloads are compute-bound.

(b) Decode. As model sizes and thus the number of required GPUs grow, the "Lite" cluster underperforms due to increased memory access intensities. The degradation is worse with GPT-3 due to it having more KV-heads resulting in proportionally longer memory-bound stages. As Lite-GPUs utilize their available shoreline for more memory bandwidth, performance improves and exceeds the current H100 cluster.

Figure 3: Results of the roofline modeling of H100 and Lite-H100 clusters. Note that Lite-H100 is already expected to be cheaper to manufacture, so we expect comparable performance to suffice.

the first new token. The prefill phase is usually highly parallelizable and efficient in utilizing the compute resources. The decode phase generates output tokens one at a time, with each new token building on the entire KV cache and appending to it. This phase is often memory-bound and less efficient in compute utilization. In the evaluation, we assume that different phases can execute on different Lite-GPU clusters [40, 63] to demonstrate the hardware benefits achievable with Lite-GPUs. With our case study on serving latest generative AI workloads, we aim to highlight potential advantages of Lite-GPUs that are modified from today's leading GPUs.

Methodology and workload We use roofline modeling [57] to capture important hardware and software characteristics and to model a Lite-GPU cluster running LLM inference. We model important metrics including FLOPS, memory accesses, and the network traffic of collectives. The modeling measures compute stages individually, including projection, MLP, and fused FlashAttention [43]. Compute, memory I/O, and network I/O can overlap within each stage and tensor parallelism is used to distribute execution within each cluster.

NVIDIA H100 is the baseline GPU for comparison [11]. An H100 cluster consists of one to eight H100 GPUs. Each H100 includes 132 SMs. The Lite-GPU is modeled based on H100 by reducing its capabilities to 1/4 of the original, denoted as "Lite" in Table 1. Accordingly, a Lite-H100 cluster can consist of one to 32 Lite-GPUs, to match the total maximum number of SMs of the H100 cluster. Recall that for Lite-H100, we expect that bandwidth-to-compute can increase to 2× of H100 and that it can deliver higher sustainable FLOPS due to improved cooling efficiency. To explore how these hardware improvements can impact performance, we further define customized Lite-GPUs for comparison, as denoted and summarized in Table 1, with changed parameters highlighted in blue and red.

Table 1: GPU configurations

GPU type	TFLOPS	Cap. GB	Mem BW GB/s	Net BW GB/s	#Max GPUs
H100	2000	80	3352	450	8
Lite	500	20	838	112.5	32
Lite+NetBW	500	20	838	225	32
Lite+NetBW+FLOPS	550	20	419	225	32
Lite+MemBW	500	20	1675	112.5	32
Lite+MemBW+NetBW	500	20	1675	225	32

We evaluate performance with three LLM models with different sizes and structures: Llama3-70B, GPT3-175B, and Llama3-405B [7, 32]. We define the search criteria based on Splitwise's latency requirements, with TTFT (time-to-first-token) ≤ 1s and TBT (time-between-tokens) ≤ 50ms constraints [40]. We set a constant prompt sequence length of 1500 tokens, the reported median size in a production workload for coding [40]. The search sweeps all possible batch sizes and number of GPUs for each GPU type. Then, since different GPU types have different hardware capabilities, we normalize the throughput for each configuration using the number of SMs in that configuration. The resulting metric, throughput per SM (tokens/s/SM), represents the *performance efficiency* of that configuration. For each GPU type, we plot the configuration with the highest throughput per SM. Note that while we sweep up to the maximum number of GPUs per cluster as defined in Table 1, the search may return that running a model with less GPUs than the maximum yields better throughput per SM.

Results The results are summarized in Figure 3. With this study, we show that while the basic Lite-GPU with no additional networking support could face performance limitations, a Lite-GPU cluster can be customized to match or improve on the performance of a typical H100 cluster. Note that this is in addition to the hardware and systems advantages of Lite-GPUs described in previous sections. Additionally, note that customized and improved Lite-GPUs need not consume more energy at the cluster level, as, e.g., they can trade-off FLOPS for bandwidth.

In terms of performance per $-cost, which is the primary metric for cloud operators, we expect the cost per comparable deployments to decrease with Lite-GPU, due to the improvements in the manufacturing cost of GPUs. In this case, even matching performance of today's clusters may lead to sufficient improvement in performance per cost. Nevertheless, the additional cost of networking needs consideration, and while it may be initially a fraction of the GPU cost, it can turn into a bottleneck with increased scale. Further analysis on performance and total cost of operation is vital for the viability of deploying Lite-GPUs at scale, though it is out-of-scope for this paper.

5 RELATED WORK

Running AI workloads on small chips has gained traction in the past years. For example, Apple has been shipping Neural Engine in their mobile devices since 2017 [54]. Most recently, NVIDIA announced DIGITS as a powerful GPU workstation for engineering AI models prior to deployment on the cloud [39]. Also from the model design direction, improving inference for single GPUs has gained significant research attention [3, 45, 58–60]. While these efforts aim to maximize AI capabilities on a single device, they do not address the challenges of scaling demanding AI workloads in the data-center.

On the other hand, Google's TPUs are an example of scaling AI workloads across many tensor processors [24]. While they employ advanced networking technologies for lower cost and power consumption, performance and flexibility limitations remain, such as a long reconfiguration periods and multi-device blast radii, due to which a failure can render a group of TPUs inactive. TPUs share similar principles with Lite-GPUs. However, TPUs are specialized and offer less programming flexibility compared to GPUs. Additionally, TPUs have also packed more transistors into the same package across generations and are on a similar path to current complex GPUs [10, 24].

Alternatively to the scale-out approach of Lite-GPUs, wafer-scale computing systems aim to pack massive amounts of compute and communication bandwidth onto single, large integrated chips [8, 23]. Though these systems benefit from much increased bandwidth and integration density, they require complex and advanced packaging techniques, which can lead to challenges around yield, cost, and power consumption [23].

There is a plethora of work that propose systems solutions for improving the performance [4, 13], energy efficiency [42, 46], parallelism [27, 44], and scheduling [17, 49] of AI workloads in the data-center. Recently, DeepSeek demonstrated a variety of optimizations that enable efficient training and serving of a strong LLM on hardware that are relatively weaker than cutting-edge GPUs [14]. These works are complementary to the hardware and systems efforts on delivering cost-effective scaling of AI workloads using Lite-GPUs.

6 CONCLUSION

We are already facing uncertainty on the amount of compute and memory that can fit into a single GPU package, as cutting-edge GPUs already display the packaging, cooling, power- and cost-related challenges of their complex designs. In this paper, we propose an alternative way of scaling AI infrastructure: by using Lite-GPUs instead of complex and expensive large GPUs. Motivated by the yield, power, and operational benefits of smaller GPU packages, we look at AI infrastructure within the context of Lite-GPUs. We provide an overview of key research questions around workload, memory, and network management. We also present how Lite-GPUs can improve energy management, performance efficiency, and fault-tolerance. With this paper, we aim to start a discussion around Lite-GPUs and their potential to turn the tide on the many issues we face while building and operating GPU clusters in the era of generative AI.

REFERENCES

[1] [n. d.]. NVIDIA GB200 NVL72. https://www.nvidia.com/en-us/data-center/gb200-nvl72/. Accessed: 2025-01-15.

[2] 2024. Hot Chips 2024 Conference Proceedings. https://hc2024.hotchips.org/assets/program/conference/day1/HotChips%20-%202024-08-26.pdf Accessed: 2025-01-16.

[3] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel

Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv:2404.14219 [cs.CL] https://arxiv.org/abs/2404.14219

[4] Amey Agrawal, Ashish Panwar, Jayashree Mohan, Nipun Kwatra, Bhargav S. Gulavani, and Ramachandran Ramjee. 2023. SARATHI: Efficient LLM Inference by Piggybacking Decodes with Chunked Prefills. arXiv:2308.16369 [cs.LG] https://arxiv.org/abs/2308.16369

[5] Reza Yazdani Aminabadi, Samyam Rajbhandari, Minjia Zhang, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Jeff Rasley, Shaden Smith, Olatunji Ruwase, and Yuxiong He. 2022. DeepSpeed Inference: Enabling Efficient Inference of Transformer Models at Unprecedented Scale. arXiv:2207.00032 [cs.LG] https://arxiv.org/abs/2207.00032

[6] Hitesh Ballani, Paolo Costa, Raphael Behrendt, Daniel Cletheroe, Istvan Haller, Krzysztof Jozwik, Fotini Karinou, Sophie Lange, Kai Shi, Benn Thomsen, and Hugh Williams. 2020. Sirius: A Flat Datacenter Network with Nanosecond Optical Switching. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication* (Virtual Event, USA) *(SIGCOMM '20)*. Association for Computing Machinery, New York, NY, USA, 782–797. doi:10.1145/3387514.3406221

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL] https://arxiv.org/abs/2005.14165

[8] Cerebras. [n. d.]. Wafer Scale Engine 3. https://www.cerebras.ai/chip. [Accessed 17-04-2025].

[9] Jae-Won Chung, Yile Gu, Insu Jang, Luoxi Meng, Nikhil Bansal, and Mosharaf Chowdhury. 2024. Reducing Energy Bloat in Large Model Training. In *Proceedings of the ACM SIGOPS 30th Symposium on Operating Systems Principles* (Austin, TX, USA) *(SOSP '24)*. Association for Computing Machinery, New York, NY, USA, 144–159. doi:10.1145/3694715.3695970

[10] Google Cloud. 2025. TPU v6e. https://cloud.google.com/tpu/docs/v6e Accessed: 2025-01-15.

[11] NVIDIA Corporation. 2022. NVIDIA H100 Tensor Core GPU Architecture Overview. https://resources.nvidia.com/en-us-tensor-core/gtc22-whitepaper-hopper Accessed: 2025-01-10.

[12] NVIDIA Corporation. 2025. NVIDIA H100 NVL GPU. https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/h100/PB-11773-001_v01.pdf Accessed: 2025-01-15.

[13] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems* 35 (2022), 16344–16359.

[14] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2025. DeepSeek-V3 Technical Report. arXiv:2412.19437 [cs.CL] https://arxiv.org/abs/2412.19437

[15] Rob Van der Wijngaart and Fred Oh. 2022. Boosting Application Performance with GPU Memory Prefetching. https://developer.nvidia.com/blog/boosting-application-performance-with-gpu-memory-prefetching/ Accessed: 2025-01-15.

[16] Hugging Face. [n. d.]. Llama 3.1 - 405B, 70B & 8B with multilinguality and long context. https://huggingface.co/blog/llama31. [Accessed 15-04-2025].

[17] Yichao Fu, Siqi Zhu, Runlong Su, Aurick Qiao, Ion Stoica, and Hao Zhang. 2024. Efficient LLM Scheduling by Learning to Rank. arXiv:2408.15792 [cs.LG] https://arxiv.org/abs/2408.15792

[18] Phil Garrou. 2024. IFTLE 607: Why Nvidia's Blackwell is Having Issues with TSMC CoWoS-L Technology. *3DInCites* (2024). https://www.3dincites.com/2024/10/iftle-607-why-nvidias-blackwell-is-having-issues-with-tsmc-cowos-l-technology/ Accessed: 2025-01-14.

[19] A. Gupta and J.W. Lathrop. 1972. Yield analysis of large integrated-circuit chips. *IEEE Journal of Solid-State Circuits* 7, 5 (1972), 389–395. doi:10.1109/JSSC.1972.1052898

[20] Horace He. 2024. Strangely, Matrix Multiplications on GPUs Run Faster When Given "Predictable" Data! https://www.thonking.ai/p/strangely-matrix-multiplications Accessed: 2025-01-15.

[21] Ali Heydari, Pardeep Shahi, Vahideh Radmard, Bahareh Eslami, Uschas Chowdhury, Satyam Saini, Pratik Bansode, Harold Miyamura, Dereje

Agonafer, and Jeremy Rodriguez. 2022. Liquid to Liquid Cooling for High Heat Density Liquid Cooled Data Centers. In *International Electronic Packaging Technical Conference and Exhibition*, Vol. 86557. American Society of Mechanical Engineers, V001T01A007.

[22] Freddie Hong, Iason Sarantopoulos, Elliott Hogg, David Richardson, Yizhong Zhang, Hugh Williams, David Sweeney, Andromachi Chatzieleftheriou, and Antony Rowstron. 2024. Self-maintaining [networked] systems: The rise of datacenter robotics!. In *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks* (Irvine, CA, USA) *(HotNets '24)*. Association for Computing Machinery, New York, NY, USA, 159–166. doi:10.1145/3696348.3696872

[23] Yang Hu, Xinhan Lin, Huizheng Wang, Zhen He, Xingmao Yu, Jiahao Zhang, Qize Yang, Zheng Xu, Sihan Guan, Jiahao Fang, Haoran Shang, Xinru Tang, Xu Dai, Shaojun Wei, and Shouyi Yin. 2024. Wafer-Scale Computing: Advancements, Challenges, and Future Perspectives [Feature]. *IEEE Circuits and Systems Magazine* 24, 1 (2024), 52–81. doi:10.1109/MCAS.2024.3349669

[24] Norm Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, Clifford Young, Xiang Zhou, Zongwei Zhou, and David A Patterson. 2023. TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings. In *Proceedings of the 50th Annual International Symposium on Computer Architecture* (Orlando, FL, USA) *(ISCA '23)*. Association for Computing Machinery, New York, NY, USA, Article 82, 14 pages. doi:10.1145/3579371.3589350

[25] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. arXiv:2309.06180 [cs.LG] https://arxiv.org/abs/2309.06180

[26] Heting Liu, Zhichao Li, Cheng Tan, Rongqiu Yang, Guohong Cao, Zherui Liu, and Chuanxiong Guo. 2023. Predicting GPU Failures With High Precision Under Deep Learning Workloads. In *Proceedings of the 16th ACM International Conference on Systems and Storage*. 124–135.

[27] Hao Liu, Matei Zaharia, and Pieter Abbeel. 2023. Ring Attention with Blockwise Transformers for Near-Infinite Context. arXiv:2310.01889 [cs.CL] https://arxiv.org/abs/2310.01889

[28] Mark Liu and H.-S. Philip Wong. 2024. How We'll Reach a 1 Trillion Transistor GPU. *IEEE Spectrum* (March 2024). https://spectrum.ieee.org/trillion-transistor-gpu Accessed: 2025-01-15.

[29] Gabriel H Loh, Samuel Naffziger, and Kevin Lepak. 2021. Understanding chiplets today to anticipate future integration opportunities and limits. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 142–145.

[30] Hasan Al Maruf and Mosharaf Chowdhury. 2023. Memory Disaggregation: Advances and Open Challenges. arXiv:2305.03943 [cs.DC] https://arxiv.org/abs/2305.03943

[31] Meta. [n. d.]. Introducing Llama 3.1: Our most capable models to date. https://ai.meta.com/blog/meta-llama-3-1/. [Accessed 15-04-2025].

[32] Meta. 2025. Meta Llama on Hugging Face. https://huggingface.co/meta-llama. Accessed: 2025-01-13.

[33] Xupeng Miao, Chunan Shi, Jiangfei Duan, Xiaoli Xi, Dahua Lin, Bin Cui, and Zhihao Jia. 2024. SpotServe: Serving Generative Large Language Models on Preemptible Instances. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2* (La Jolla, CA, USA) *(ASPLOS '24)*. Association for Computing Machinery, New York, NY, USA, 1112–1127. doi:10.1145/3620665.3640411

[34] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large Language Models: A Survey. arXiv:2402.06196 [cs.CL] https://arxiv.org/abs/2402.06196

[35] Cyriel Minkenberg, Rajagopal Krishnaswamy, Aaron Zilkie, and David Nelson. 2021. Co-packaged datacenter optics: Opportunities and challenges. *IET optoelectronics* 15, 2 (2021), 77–91.

[36] Moore Elite. [n. d.]. Die Yield Calculator. http://cloud.mooreelite.com/tools/die-yield-calculator/index.html. Accessed: 2025-01-15.

[37] Jowi Morales. 2024. Nvidia Blackwell GPUs Allegedly Delayed Due to Design Flaws. *Tom's Hardware* (2024). https://www.tomshardware.com/pc-components/gpus/nvidia-blackwell-gpus-allegedly-delayed-due-to-design-flaws Accessed: 2025-01-10.

[38] NVIDIA. [n. d.]. NVIDIA Announces Spectrum-X Photonics, Co-Packaged Optics Networking Switches to Scale AI Factories to Millions of GPUs. https://nvidianews.nvidia.com/news/nvidia-spectrum-x-co-packaged-optics-networking-switches-ai-factories. [Accessed 15-04-2025].

[39] NVIDIA. 2025. NVIDIA Project DIGITS. https://www.nvidia.com/en-us/project-digits/. Accessed: 2025-01-14.

[40] Pratyush Patel, Esha Choukse, Chaojie Zhang, Aashaka Shah, Íñigo Goiri, Saeed Maleki, and Ricardo Bianchini. 2024. Splitwise: Efficient generative llm inference using phase splitting. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 118–132.

[41] Leonardo Piga, Iyswarya Narayanan, Aditya Sundarrajan, Matt Skach, Qingyuan Deng, Biswadip Maity, Manoj Chakkaravarthy, Alison Huang, Abhishek Dhanotia, and Parth Malani. 2024. Expanding Datacenter Capacity with DVFS Boosting: A safe and scalable deployment experience. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1* (La Jolla, CA, USA) *(ASPLOS '24)*. Association for Computing Machinery, New York, NY, USA, 150–165. doi:10.1145/3617232.3624853

[42] Haoran Qiu, Weichao Mao, Archit Patke, Shengkun Cui, Saurabh Jha, Chen Wang, Hubertus Franke, Zbigniew Kalbarczyk, Tamer Başar, and Ravishankar K Iyer. 2024. Power-aware Deep Learning Model Serving with {μ-Serve}. In *2024 USENIX Annual Technical Conference (USENIX ATC 24)*. 75–93.

[43] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. 2024. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *arXiv preprint arXiv:2407.08608* (2024).

[44] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2020. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. arXiv:1909.08053 [cs.CL] https://arxiv.org/abs/1909.08053

[45] Yixin Song, Zeyu Mi, Haotong Xie, and Haibo Chen. 2024. Powerinfer: Fast large language model serving with a consumer-grade gpu. In *Proceedings of the ACM SIGOPS 30th Symposium on Operating Systems Principles*. 590–606.

[46] Jovan Stojkovic, Esha Choukse, Chaojie Zhang, Inigo Goiri, and Josep Torrellas. 2024. Towards Greener LLMs: Bringing Energy-Efficiency to the Forefront of LLM Inference. *arXiv preprint arXiv:2403.20306* (2024).

[47] Jovan Stojkovic, Pulkit A. Misra, Íñigo Goiri, Sam Whitlock, Esha Choukse, Mayukh Das, Chetan Bansal, Jason Lee, Zoey Sun, Haoran Qiu, Reed Zimmermann, Savyasachi Samal, Brijesh Warrier, Ashish Raniwala, and Ricardo Bianchini. 2024. SmartOClock: Workload- and Risk-Aware Overclocking in the Cloud. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*. 437–451. doi:10.1109/ISCA59077.2024.00040

[48] Foteini Strati, Sara Mcallister, Amar Phanishayee, Jakub Tarnawski, and Ana Klimovic. 2024. DéjàVu: KV-cache Streaming for Fast, Fault-tolerant Generative LLM Serving. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*, Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.). PMLR, 46745–46771. https://proceedings.mlr.press/v235/strati24a.html

[49] Biao Sun, Ziming Huang, Hanyu Zhao, Wencong Xiao, Xinyi Zhang, Yong Li, and Wei Lin. 2024. Llumnix: Dynamic Scheduling for Large Language Model Serving. *arXiv preprint arXiv:2406.03243* (2024).

[50] Min Tan, Jiang Xu, Siyang Liu, Junbo Feng, Hua Zhang, Chaonan Yao, Shixi Chen, Hangyu Guo, Gengshi Han, Zhanhao Wen, et al. 2023. Co-packaged optics (CPO): status, challenges, and solutions. *Frontiers of Optoelectronics* 16, 1 (2023), 1.

[51] Tianqi Tang and Yuan Xie. 2022. Cost-Aware Exploration for Chiplet-Based Architecture with Advanced Packaging Technologies. arXiv:2206.07308 [cs.AR] https://arxiv.org/abs/2206.07308

[52] DW Team. 2025. Nvidia faces order delays as Blackwell chips overheat. *digwatch* (2025). https://dig.watch/updates/nvidia-faces-order-delays-as-blackwell-chips-overheat Accessed: 2025-01-14.

[53] D. Teets. 1996. A model for radial yield degradation as a function of chip size. *IEEE Transactions on Semiconductor Manufacturing* 9, 3 (1996), 467–471. doi:10.1109/66.536118

[54] The Mac Observer. 2025. What is Apple Neural Engine? https://www.macobserver.com/tips/what-is-apple-neural-engine/ Accessed: 2025-01-14.

[55] Ajay Tirumala and Raymond Wong. 2024. NVIDIA Blackwell Platform: Advancing Generative AI and Accelerated Computing. In *2024 IEEE Hot Chips 36 Symposium (HCS)*. IEEE Computer Society, 1–33.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. arXiv:1706.03762 [cs.CL] https://arxiv.org/abs/1706.03762

[57] Samuel Williams, Andrew Waterman, and David Patterson. 2009. Roofline: an insightful visual performance model for multicore architectures. *Commun. ACM* 52, 4 (April 2009), 65–76. doi:10.1145/1498765.1498785

[58] Daliang Xu, Hao Zhang, Liming Yang, Ruiqi Liu, Gang Huang, Mengwei Xu, and Xuanzhe Liu. 2024. Empowering 1000 tokens/second on-device llm prefilling with mllm-npu. *arXiv preprint arXiv:2407.05858* (2024).

[59] Daliang Xu, Hao Zhang, Liming Yang, Ruiqi Liu, Gang Huang, Mengwei Xu, and Xuanzhe Liu. 2024. Fast On-device LLM Inference with NPUs. arXiv:2407.05858 [cs.AI] https://arxiv.org/abs/2407.05858

[60] Jiajun Xu, Zhiyuan Li, Wei Chen, Qun Wang, Xin Gao, Qi Cai, and Ziyuan Ling. 2024. On-Device Language Models: A Comprehensive Review. arXiv:2409.00088 [cs.CL] https://arxiv.org/abs/2409.00088

[61] Weizheng Xu, Youtao Zhang, and Xulong Tang. 2021. Parallelizing DNN Training on GPUs: Challenges and Opportunities. In *Companion Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 174–178. doi:10.1145/3442442.3452055

[62] Junwen Zhang and Zhensheng Jia. 2022. Coherent Passive Optical Networks for 100G/λ-and-Beyond Fiber Access: Recent Progress and Outlook. *IEEE Network* 36, 2 (2022), 116–123. doi:10.1109/MNET.005.2100604

[63] Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. 2024. DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving. arXiv:2401.09670 [cs.DC] https://arxiv.org/abs/2401.09670