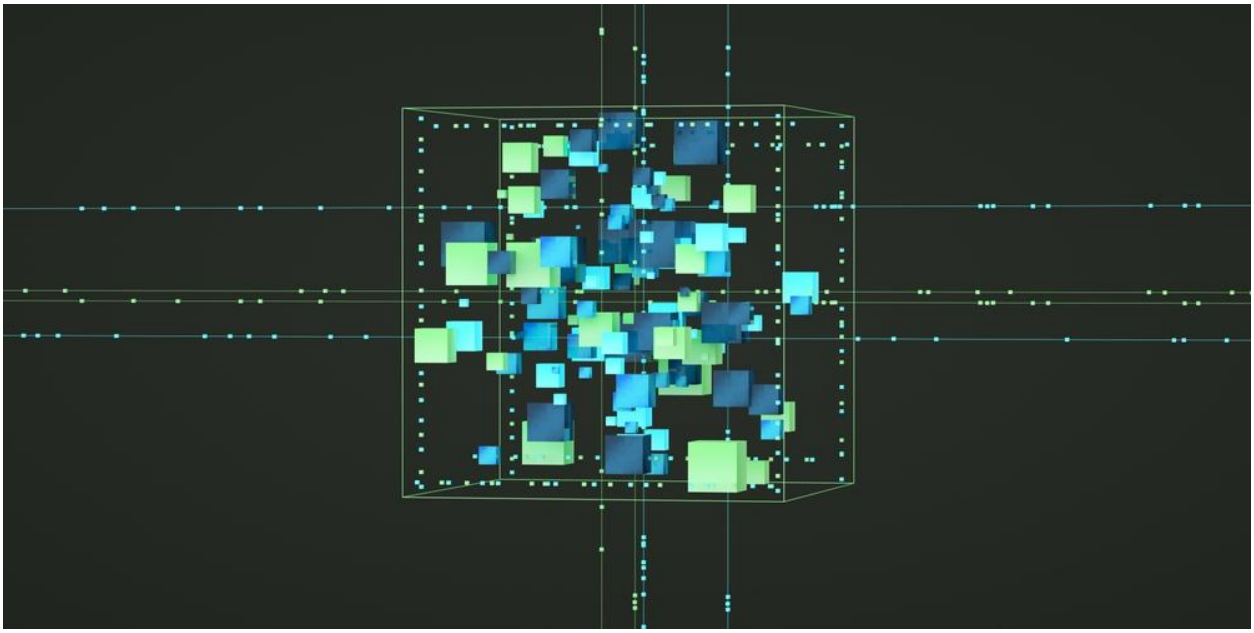

GPUs are not a Good Architectural Choice for Processing Modern LLMs

Lauro Rizzatti
Business Development Executive
VSORA

There's a quiet irony in today's AI world. GPUs, the very chips that made the deep learning revolution possible are beginning to look less suited for the newest generation of large language models (LLMs) like GPT-4 and GPT-5. They are still powerful, still essential, and still widely used. Architecturally, they are no longer a natural match for how modern LLMs work.

Why SIMT-based GPUs struggle with modern LLMs

To understand why, it helps to picture how a GPU operates. Most GPUs are built around what's called the SIMT model—Single Instruction, Multiple Threads. In simple terms, imagine thousands of disciplined soldiers standing in formation. You shout one command—"Multiply!"—and all of them perform the exact same operation at the same time. This is perfect for graphics rendering and early neural networks, where the same mathematical operation is repeated again and again across large blocks of data. Uniformity works. Predictable patterns. Everyone's busy.

Modern language models, however, don't behave like a repetitive math drill. They operate less like an army of soldiers and more like a symphonic orchestra. Each player follows their own intricate score, contributing to a sound that is restless, majestic, and constantly shifting in a state of dynamic evolution. Instead of treating every piece of data the same way, LLMs rely heavily on mechanisms such as attention and Mixture of Experts (MoE). In these systems, only certain parts of the model "wake up" depending on the word, the sentence, or the context being processed.

Now the mismatch becomes clear. GPUs want all their soldiers performing the same task at once. An MoE model might activate only a small fraction of its internal "experts" for a given word. If 100 compute units have work to do and 4,900 don't, the rest sit there waiting. From a

hardware perspective, it's like paying for a full orchestra and only allowing the violin section to play.

The deleterious impact of the "memory wall" on processing LLMs on GPUs
A more profound bottleneck is the memory wall, the growing performance gap where processors can execute arithmetic operations far faster than memory systems can supply data, causing increasingly powerful compute units to sit idle while waiting on bandwidth- and latency-limited data movement.

With models containing hundreds of billions—or even trillions—of parameters, the real bottleneck isn't raw computation anymore. GPUs can perform math extraordinarily fast. The problem is feeding them data quickly enough. You can think of it as the difference between thinking and reading. The GPU can "think" at incredible speed, but it struggles to "read" the next piece of information from memory fast enough to stay busy.

Data has to travel from high-bandwidth memory into compute units and back again. See Table 1.

Capacity	Energy Dissip.		Bandwidth	Latency
100 B – 10 kB	~0.1-0.5 pJoules	Reg.	20-100 TB/sec	~1 cycle
32 kB – 256 kB	~1-5 pJoules	Shared	10-100 TB/s	~1-3 cycles
16 kB – 256 kB	~5-10 pJoules	L1 Cache	1-10 TB/s	~3-10 cycles
256 kB – 40+ MB	~20-50 pJoules	L2 Cache	0.5-5 TB/s	~5-50 cycles
8 GB – 192 GB	~0.1-1 nJoules	HBM3	0.1-1.2 TB/s	~100-300 cycles

Table #1 caption: Comparison of capacity, energy consumption, bandwidth and latency in a tpycal memory hierarcgy.
Source: Author

With massive models, that pathway becomes permanently congested. Increasingly, more energy is spent moving data around than performing calculations. The chip's arithmetic units wait idly for data to arrive. And because SIMT requires coordination across thousands of threads, there is additional internal bookkeeping overhead layered on top of the memory bottleneck.

The mismatch between MoE and GPU's processing
The situation becomes even more complex with Mixture of Experts. Imagine running a large translation bureau with 1,000 specialists. In older "dense" models, every time a word arrives, all 1,000 specialists examine it together. That fits the GPU model perfectly: one instruction, everyone executes. In an MoE system, the bureau is divided into highly specialized teams. If the word "subpoena" arrives, only the legal experts are activated. The rest remain idle. From a software efficiency standpoint, this selective activation is brilliant.

From a SIMT hardware perspective, it's uncomfortable. The hardware wants uniformity; the model delivers conditional branching and sparsity.

Conditional branching introduces another headache. If different threads follow different logical paths, the GPU often has to execute those paths sequentially while masking out irrelevant results. In effect, what should have been parallel work becomes serialized. Performance drops, power efficiency suffers, and utilization falls.

Then there's the communication tax. Modern LLMs are so large they cannot fit on a single chip. They are distributed across many GPUs, sometimes thousands. If the routing logic determines that a particular "expert" lives on another chip, data must travel across the data center network. GPUs are excellent at talking to their own memory.

They are less efficient at shouting across a server rack to coordinate specialized fragments of a model. While data moves, expensive hardware sits waiting. It's not uncommon for utilization levels to hover around 10 or less percent in certain workloads, meaning the system is consuming nearly full power while doing a fraction of its potential work.

Alternative AI processing architectures are mandatory

This is why the industry is gradually exploring alternatives. Newer AI processors are being designed around dataflow principles rather than SIMT's army-style coordination. Instead of thousands of units waiting for a shared command, data flows through purpose-built compute structures more like an assembly line. The emphasis shifts from thread management to efficient movement of information. The hardware is designed specifically around matrix operations, attention mechanisms, and predictable data streaming.

In simple terms, using a SIMT GPU for a trillion-parameter language model is a bit like using a high-performance sports car for parcel delivery. It's incredibly fast in short bursts, but it's not optimized for constant stops, heavy loads, or energy efficiency at scale. A dedicated logistics system might be less glamorous, but it's built for exactly that job.

Action	Older Dense AI	Modern MoE LLMs
Workflow	Everyone works on everything	Only selected experts activate
GPU usage	Fully occupied	Lots of idle time
Data movement	Predictable	Scattered and irregular

Table #2 caption: A simplified chart illustrates the efficiency gap.
Source: Author

Conclusions

None of this means GPUs are obsolete. They remain foundational to AI and will continue to play a role. In fact, they are currently irreplaceable for the training phase of modern LLMs where massive FLOPS count more than latency, efficiency and cost. As language models become larger, more conditional, more memory-heavy, and more distributed, the architectural tension becomes harder to ignore.

GPUs launched the AI era. The next phase may belong to hardware designed not around uniform parallelism, but around how modern language models think—selectively, sparsely, and with enormous data movement at their core.

About Lauro Rizzatti
Lauro Rizzatti is a business development executive with VSORA, a pioneering technology company offering silicon semiconductor solutions that redefine performance, and a noted chip design verification consultant and industry expert on hardware emulation.