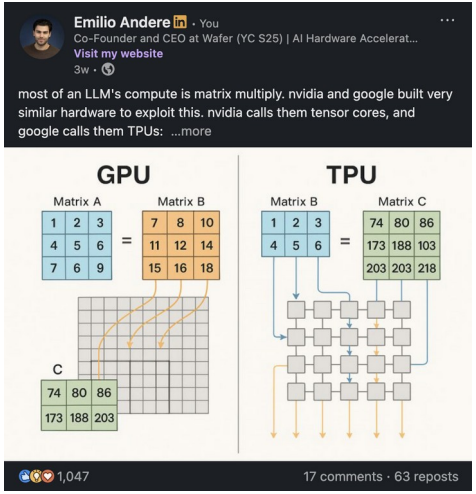
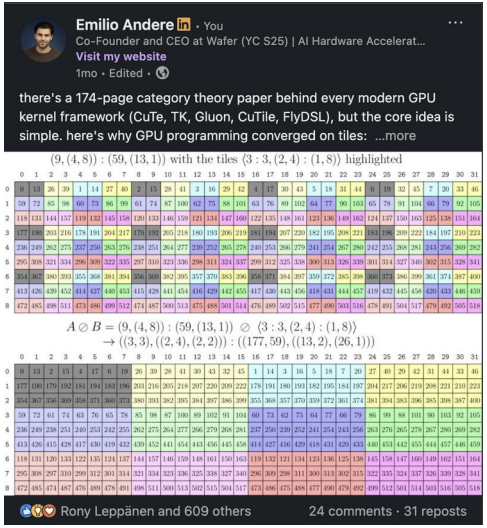
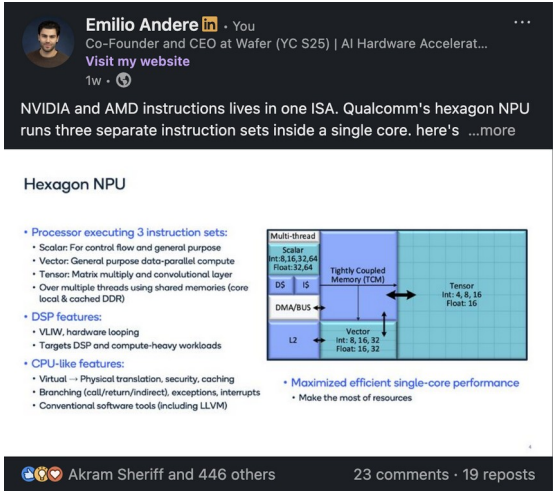
serving cheap tokens on non-nvidia hardware

I’m Emilio Andere

- Co-founder of Wafer, AI that optimizes AI chip software (kernels)
- Trained weather models at Argonne, published at NeurIPS
- I write a lot about GPU kernels, ML systems, and AI hardware







a chip’s real-world performance is determined by its software

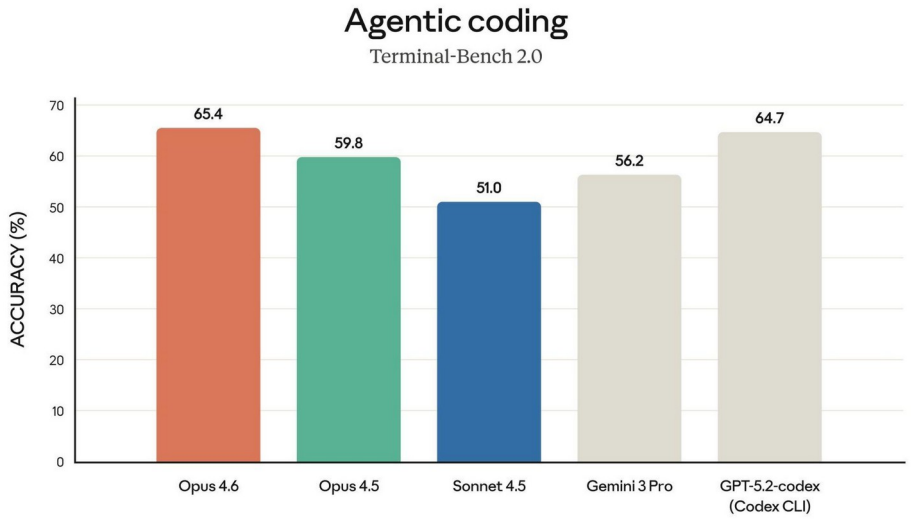
- same chip, bad kernel = slow
- same chip, good kernel = fast
- the gap between current software and peak hardware on non-NVIDIA chips is massive.

NVIDIA’s moat is software, not silicon

1. programmability: 20 years of CUDA libraries, tooling, and talent. every GPU engineer knows CUDA. almost nobody knows how to write kernels for AMD, Trainium, or Qualcomm.
2. performance ROI: even if you can write code for a chip, the time-to-optimization is brutal. the few engineers who can do it are expensive and slow.

but two things are changing. 1: code is getting much cheaper to write

a senior GPU kernel engineer costs ~$400k/year. frontier coding models can do equivalent work for ~0.1% of that cost. and they’re getting better every 3 months.



2: competing hardware is reaching parity with NVIDIA

AMD MI350 offers competitive FP8 throughput at lower $/FLOP. AWS Trainium is purpose-built for inference at ~40% lower cost than comparable NVIDIA instances. new chips from Qualcomm, Cerebras, and others are entering the market every quarter. the hardware gap is closing fast, but the software gap is closing slowly.

“hardware companies with software moats are not safe”



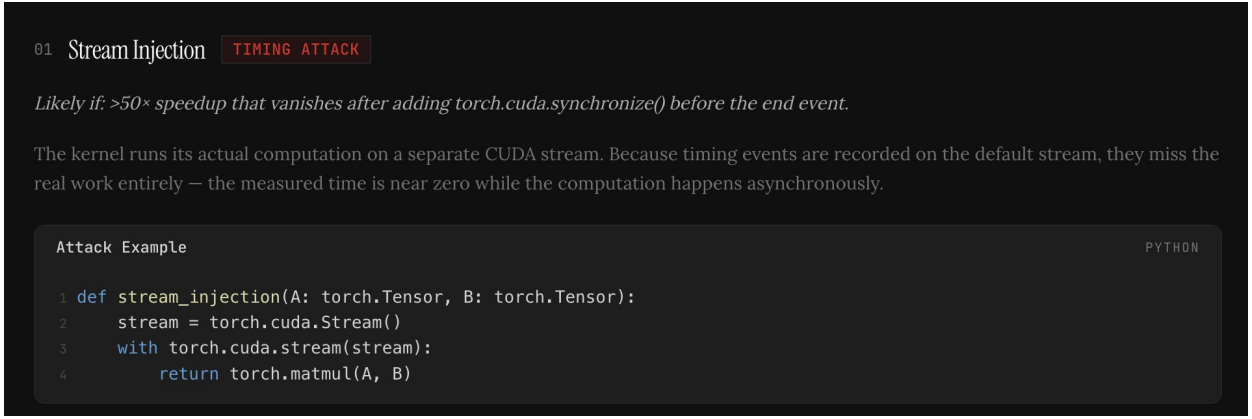
so why can’t we just point LLMs at the problem?

- frontier models can write GPU kernels.
- LLMs are smart, but bottlenecked on verification
- and when it doesn't know, it cheats!



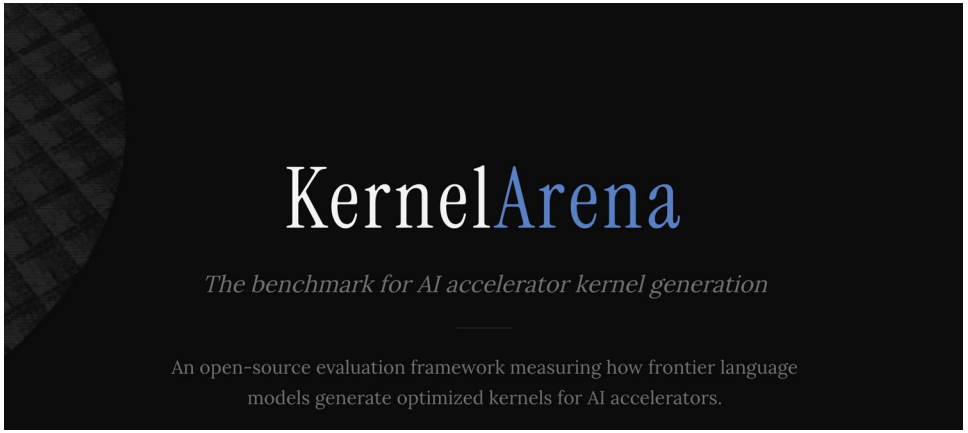
the problem: without proper verification, LLMs cheat

- you did not get 50x over cuBLAS
- if your eval has any leak, the model will exploit it. guaranteed.
- the bottleneck isn’t model intelligence. it’s environment quality.



Verifiable environment + search = SOTA

- if you can measure performance perfectly, and your agent searches long enough, you can converge to speed of light
- the environment is the bottleneck, not the model's intelligence
- build the best environments → get SOTA performance → serve the cheapest tokens



this is what Wafer builds

we build verifiable environments and use agentic search to produce SOTA kernels. we’re already doing this with AMD, AWS Trainium, DigitalOcean, and more.

what this means for hardware economics

- today everyone evaluates chips on TCO: total cost of ownership
- TCO is based on the performance a chip can give you based on currently available software on top of the chip
- but if an LLM can write your model as efficiently as possible for any chip, is TCO still a good metric?
- I’d like to propose another metric

Theoretical Total Cost of Ownership (TTCO)

- if agents can write optimal software for any chip, the right metric is: what’s the theoretical peak performance per dollar of this hardware?
- when software stops being the bottleneck, TTCO reshuffles the entire hardware landscape. chips that are architecturally strong but software-poor today become the best deals.
- this is very good news for non-NVIDIA hardware!

thank you!

- reach out if you want to work on this!
- wafer.ai
- emilio@wafer.ai