Notes from building a local AI inference system

Tommaso Bilotta

This document describes the design, construction, and validation of a fully custom supercomputing system built for sustained local AI inference. It covers the complete engineering journey: initial failures, know-how acquisition across multiple disciplines, hardware selection challenges, iterative design through five revisions, and final validation with 12-hour benchmarks. All data is included.

System: 6 NVIDIA Tesla K80 cards (12 logical GPUs), custom aluminum and 3D-printed structure designed in CAD, validated under 12-hour sustained load at 100% GPU utilization. Fifth hardware iteration.

Results: 8 GPUs at 100% for 12 hours. Temperature range 35–52°C (vs. 75–90°C in standard server). Zero thermal drift. Conversational acoustic levels at 1 meter. No datacenter cooling. Residential environment.

Notes from building a local AI inference system

My first local inference test lasted 180 seconds.

Four NVIDIA K80 GPUs in a standard tower server, running GPT-J. At 85°C, thermal throttling kicked in. At 90°C, all four cards shut down.

I added aftermarket fans. The delta was about 5°C — thermally irrelevant under sustained compute.

That result eliminated the assumption that inference is a software problem you solve with a config file.

> **Engineering context**
> The NVIDIA K80 (Kepler, GK210) has no onboard cooling — it relies entirely on chassis airflow. Its thermal design power is 300W per card (150W per GPU die). In a standard tower server, the airflow is designed for CPUs and memory, not for sustained GPU compute at maximum TDP. The 90°C threshold is consistent with the K80's thermal shutdown specification under inadequate cooling.

What I didn't know at the start

I came from a software and systems background. I understood Linux, networking, orchestration. I did not understand thermal dynamics, mechanical stress, or material behavior under sustained load.

This project forced me to learn all of it. Not in theory — through failure. Every design mistake had a physical consequence: a card that shut down, a component that shifted, a motherboard that refused to POST with more than two GPUs installed.

The know-how I built over five hardware iterations is not software knowledge. It is integrated systems knowledge — the kind that spans electronics, thermodynamics, mechanical engineering, and cost management. None of it was obvious at the start. All of it turned out to be necessary.

> **On integrated knowledge**
> This trajectory — from software expertise to multi-domain systems engineering — is rarely discussed in the AI space, where the dominant narrative centers on models and frameworks. In practice, anyone attempting to run sustained AI workloads on physical hardware will encounter the same sequence of problems: thermal, mechanical, electrical, economic. The value of this documentation lies in making that trajectory explicit and empirically grounded.

Mining rigs and inference are not the same problem

The first approach was the obvious one: repurpose mining hardware. Multiple GPUs, continuous operation, high utilization. On paper, same workload.

In practice, fundamentally different.

In mining, a thermal fluctuation slows you down. In inference, it terminates the process. In mining, an interrupted hash gets recalculated. In inference, the execution is lost. Mining averages its output over time and tolerates variance. Inference requires uninterrupted continuity at 100% utilization.

I tested mining chassis under inference load. Thermal instability appeared within the first hour. The airflow geometry was engineered for a workload that absorbs fluctuation. Inference absorbs nothing.

Two configurations that behave identically at startup diverge completely under sustained stress. That distinction is invisible on any spec sheet. You discover it after the first failure.

> **Why the distinction matters**
> Mining workloads (e.g. Ethash) are inherently parallel and stateless at the hash level. A dropped computation has zero downstream impact. LLM inference, by contrast, maintains state across the entire sequence generation. A thermal event mid-inference does not just delay the output — it invalidates it. This difference in fault tolerance is the fundamental reason mining hardware cannot be repurposed for inference without redesign.

The system has nothing in common with a server

After the mining failures I stopped adapting existing hardware. I started designing from the workload backwards.

The requirement was specific: GPUs at 100% utilization, for 12 continuous hours, with no thermal shutdown, no throttling, no instability. In a domestic environment. Without external cooling infrastructure.

The result is fully custom. Structure designed in CAD. Aluminum parts cut and shaped to specification. 3D printed components where the geometry demanded it. Cable paths routed to follow airflow, not convenience. Every fastener, every material choice, every mounting angle is load-driven.

No stock chassis. No rack. The form factor exists because the thermal and mechanical constraints required exactly that shape and nothing else.

> **Design approach**
> This methodology — deriving the physical form from the workload constraints rather than adapting a workload to existing form factors — is standard practice in aerospace and automotive thermal engineering. It is almost never applied in the AI infrastructure space, where the dominant approach is to over-provision datacenter cooling rather than engineer the system itself.

Every detail is a decision

People imagine building a GPU system means choosing a case, inserting cards, and connecting cables. It does not work that way. Not at this level.

In this system, every single component is a design decision with downstream consequences. The type of screw affects how force is distributed across the mounting point. The type of washer affects vibration damping at the contact surface. The cross-section and routing of every cable affects airflow, and airflow affects temperature, and temperature affects stability.

The aluminum structure is not a chassis. It is an engineered part — designed in CAD, cut to specification, shaped to channel air in specific directions. The type of aluminum alloy matters: different alloys have different thermal conductivity, different weight, different machinability. The shape of every structural

member is load-bearing in the thermal sense, not just the mechanical one.

The 3D printed components were a chapter of their own. The material selection (rigidity, thermal resistance, dimensional stability under heat), the geometry (optimized for airflow deflection or vibration damping), the print parameters (layer adhesion, tolerances) — all of it required study and iteration. A part that worked in PLA failed under sustained heat. A part that fit at room temperature did not fit after thermal expansion. Every print run was a test, not a production step.

None of this is plug-and-play. In a system that cannot tolerate slowdowns, every component participates in the behavior of the whole. A cable routed 3 cm to the left can change the temperature of a GPU by several degrees. A mounting bolt at the wrong torque can introduce a vibration mode that amplifies over hours.

I learned this by building and breaking things, not by reading datasheets.

> **Component-level impact**
> The observation that cable routing affects GPU temperature by several degrees is consistent with thermal engineering in dense systems. In a multi-GPU chassis, the air volume is small and the heat density is high. Any obstruction in the airflow path — even a cable bundle — creates a local turbulence zone that disrupts laminar flow and reduces cooling efficiency. At 125–150W per GPU, a few degrees of temperature change represents a measurable shift in the thermal margin. This level of sensitivity is why the design required CAD-level precision rather than ad-hoc assembly.

Hardware selection: what works on paper vs what works under load

This was one of the most expensive lessons.

Choosing a motherboard for a multi-GPU system seems straightforward: check the PCIe slot count, check the power delivery, check the BIOS compatibility. I did all of that.

Several motherboards that met every specification on the datasheet were unusable in practice. Some could not POST with more than two K80 cards. Others showed instability after hours of sustained load — intermittent errors that did not appear in short tests. One board required specific BIOS settings (Above 4G Decoding, PCIe Gen2 mode) that were undocumented for this use case.

CPU selection mattered more than expected. The number of PCIe lanes, the NUMA topology, the memory bandwidth — all of these affect multi-GPU stability in ways that are invisible until you push the system to its limits.

Specification compliance and operational viability are not the same thing. That distinction cost money, not time reading documentation.

> **On hardware compatibility under stress**
> The gap between specification compliance and operational viability is a well-known problem in systems engineering. PCIe specifications define electrical and protocol compatibility, but they do not guarantee thermal stability, power delivery adequacy, or BIOS-level GPU enumeration under multi-card configurations. The Above 4G Decoding requirement is a common undocumented prerequisite for K80 cards, whose 24GB VRAM per card (12GB per die) exceeds the default 32-bit PCI memory address space. These are practical integration challenges that exist outside any specification.

Mechanical stability under sustained load

At full load, GPUs generate heat, forced airflow, and vibration. In a single desktop card, none of this matters. In a system with six K80 cards — each drawing up to 300W — packed into a dense custom structure and running for hours, all of it matters.

After several hours of continuous operation, I found micro-displacements in components that were properly secured by normal server standards. Adequate for a system that runs 20 minutes. Not adequate for 12 hours of sustained mechanical stress.

This moved the project into vibration analysis, damping strategies, and structural resonance — mechanical engineering problems, not IT problems. The type of aluminum, the geometry of the mounting, the shape of every printed component: all of it affects long-term stability. A resonance problem in the wrong place puts tens of thousands of euros of GPU hardware at risk.

The study of vibration-resistant forms for 3D printed parts became its own sub-project. GPUs under full load have extreme operational characteristics, and the structural solutions required to manage them are not trivial. The shape of a support bracket is not cosmetic — it is functional under stress, and getting it wrong has direct financial consequences.

I will not publish the specific solutions. But the problem is real, it is measurable, and it is not discussed in the AI infrastructure space.

> **On vibration in dense GPU systems**
> GPU cards under full compute load exhibit measurable vibration from multiple sources: fan rotation, VRM coil whine, and thermal expansion/contraction cycles. In a multi-card system where six cards share a structure, vibrational modes can couple. Over hours of sustained operation, this coupling can produce resonance effects invisible in short-duration testing. The fact that this was identified and addressed through structural design rather than accepted as a limitation is a significant engineering decision.

Acoustics

A system designed for domestic operation cannot sound like a datacenter. A standard rack server under GPU load typically exceeds 70 dBA — conversation in the same room becomes impossible.

I measured the acoustic profile under full load with spectral analysis. The noise floor sits between -50 and -60 dB across the spectrum, with a narrowband peak at 1680 Hz at -39 dB. The profile is stationary — flat over time, consistent with fans at steady state. No transient spikes from components cycling under variable load.

In practical terms: normal conversation was possible at one meter distance while all GPUs were running at 100%.

For a system dissipating over 1.1 kW of GPU power alone, with no liquid cooling, that acoustic profile is a direct result of airflow design and structural damping. It was a design target, not a side effect.

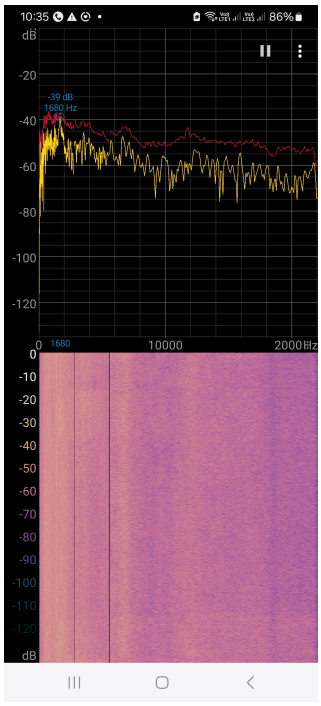

Fig. 1 — Acoustic spectral analysis under full GPU load (Spectroid). Noise floor -50 to -60 dB. Peak at 1680 Hz / -39 dB. Stationary profile.

Reading the spectral data

The stationary profile (visible in the spectrogram) confirms thermal equilibrium — fan speeds are constant because thermal load is constant. The 1680 Hz peak falls in the 1–4 kHz range where human hearing is most sensitive, consistent with a fan harmonic or VRM coil whine. The overall level permitting conversation at 1 meter suggests an estimated SPL of 50–60 dBA — roughly 15–25 dBA below a typical rack server under equivalent GPU load.

Driver constraints: 8 GPUs out of 12

The K80 is a dual-GPU card — each physical board contains two GK210 chips. The system holds six K80 cards: 12 logical GPUs total.

The NVIDIA driver stack up to CUDA 11.x, combined with the specific BIOS and platform configuration of this system, limited GPU visibility to 8 devices. The remaining four logical GPUs were physically installed but not enumerable by the driver. This is a known class of issues in dense multi-K80 configurations — dependent on BIOS PCIe aperture settings, driver version, and platform chipset. CUDA 11.8 was the last toolkit version to support Kepler (compute capability 3.7) at all. From CUDA 12 onwards, Kepler support was dropped entirely.

The 12-hour benchmarks ran with 4 physical cards — 8 GPUs active — at 100% utilization, under the maximum configuration supported by this specific platform and driver combination. Roughly 1100W of GPU power alone.

On GPU enumeration limits

The 8-GPU visibility constraint is not a universal NVIDIA driver limit. Other platforms with different chipsets, BIOS implementations, and PCIe topologies can enumerate 12, 16, or more GPUs. In this specific system, the combination of motherboard, BIOS PCIe aperture configuration, and CUDA 11.x driver stack capped enumeration at 8 devices. This is a platform-specific integration constraint, not an architectural ceiling. The two inactive K80 cards (4 GPUs) remain physically installed, powered, and contributing to thermal and mechanical load inside the chassis — but not to compute. The validated thermal envelope is therefore conservative by construction: the system manages more hardware than the benchmark exercises.

Five iterations

The current system is the fifth hardware revision.

Each version taught something the previous one could not. The first version proved that standard hardware fails under sustained inference. The second proved that mining configurations are not transferable. The third introduced custom structure and CAD-based design. The fourth addressed vibration and long-duration stability. The fifth integrated everything — thermal, mechanical, acoustic, modular — into a single validated architecture.

Each iteration was documented with benchmarks, long-duration tests, and measurement data. The progression is not theoretical. It is empirical and traceable.

Cost discipline

If the infrastructure costs more than the compute it runs, the project is dead. If the system only works with expensive hardware, the architecture is unproven.

K80s draw 300W each. They are thermally hostile, power-hungry, inefficient by any current metric. That is precisely why I started with them.

A system that survives four K80 cards at full load — 8 GPUs, 125–150W each, roughly 1100W total — in a living room in summer will handle anything newer. Upgrading to M40 or V100 class hardware means swapping cards. Nothing else changes. The GPU is a module, not a structural dependency.

The entire project was self-funded. Starting with expensive GPUs would have meant that a single design failure could have ended the experiment. Starting with K80s meant I could iterate, fail, learn, and improve without catastrophic financial risk. The economics of the project were themselves a design constraint.

Worst-case validation as methodology

If the system provides 38°C of thermal margin below throttling with K80s (the worst-case thermal scenario), then any subsequent GPU with higher efficiency per watt operates well within the validated envelope. This eliminates redesign at each GPU upgrade cycle — the structure is future-proof by construction, not by assumption.

Validation

Test conditions:

- 4 physical K80 cards active out of 6 installed — 8 GPUs, 125–150W per GPU (~1100W total)
- 100% sustained utilization across all 8 GPUs for 12 continuous hours
- Residential environment, no datacenter cooling, summer ambient temperatures
- Acoustic measurement confirming conversational noise levels at 1 meter under full load
- Spectral analysis showing stationary noise profile with no thermal cycling artifacts

No shutdowns. No throttling. No structural degradation.

GPU temperatures stabilized within minutes and held flat for the entire 12-hour window. The coolest GPU ran at 35°C. The warmest at 52°C. Average across all eight: approximately 44°C. The 17°C spread reflects position in the airflow path — but even the hottest card sat 38°C below the throttling threshold.

For reference: a K80 in a standard server under the same load runs between 75 and 90°C. The thermal graphs show zero upward drift over 12 hours. Power draw per GPU remained between 125W and 150W throughout, with no fluctuation.

The system is now in its fifth hardware iteration.

GPU	Card / Die	Temp.	Power
GPU/0	Card 1, A	~47–48°C	~135–140W
GPU/1	Card 1, B	~44–45°C	~145–148W
GPU/2	Card 2, A	~48–50°C	~128–130W
GPU/3	Card 2, B	~40–41°C	~145–148W
GPU/4	Card 3, A	~45–46°C	~128–133W
GPU/5	Card 3, B	**~35–36°C** (coolest)	~145–148W
GPU/6	Card 4, A	~50–52°C (warmest)	~125–128W
GPU/7	Card 4, B	~42–43°C	~145–148W

Table 1 — Per-GPU thermal and power data, 12h sustained benchmark. Zero drift.

Thermal distribution analysis

GPU A dies (0, 2, 4, 6) run consistently warmer than GPU B dies (1, 3, 5, 7) on the same card — consistent with GPU A being upstream in the airflow, receiving pre-heated air from its VRM section. The gradient from GPU/5 (35°C) to GPU/6 (52°C) maps the airflow topology: cards closer to the intake run cooler. Even the warmest GPU has a 38°C margin below throttling — larger than the total operating temperature of many datacenter GPUs.

12-Hour Benchmark Data — 8 GPUs at 100% Utilization

Each graph shows three metrics over ~10,000 samples (~12 hours): temperature (°C, blue), power (W, green), GPU utilization (%, red). All graphs are unaltered original data.

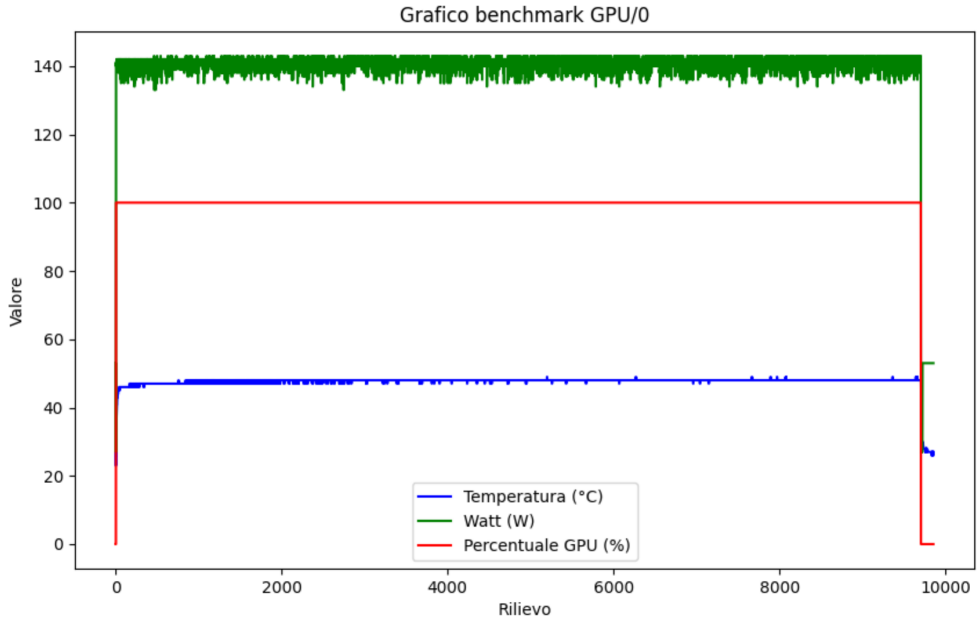

Fig. 2 — GPU/0: Card 1, A — ~47°C, ~135–140W. 12h, 100%, zero drift.

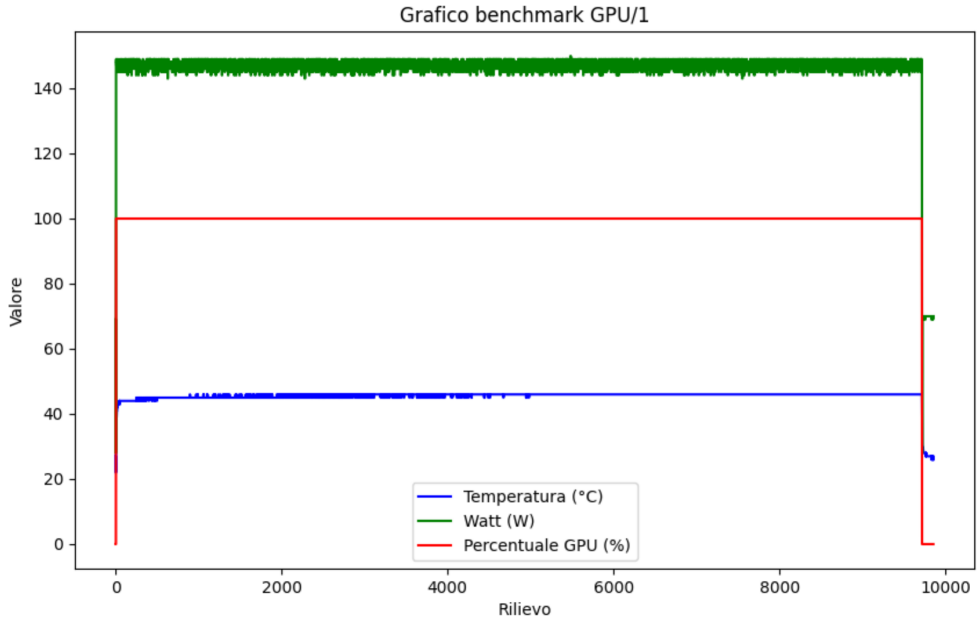

Fig. 3 — GPU/1: Card 1, B — ~44°C, ~145–148W. 12h, 100%, zero drift.

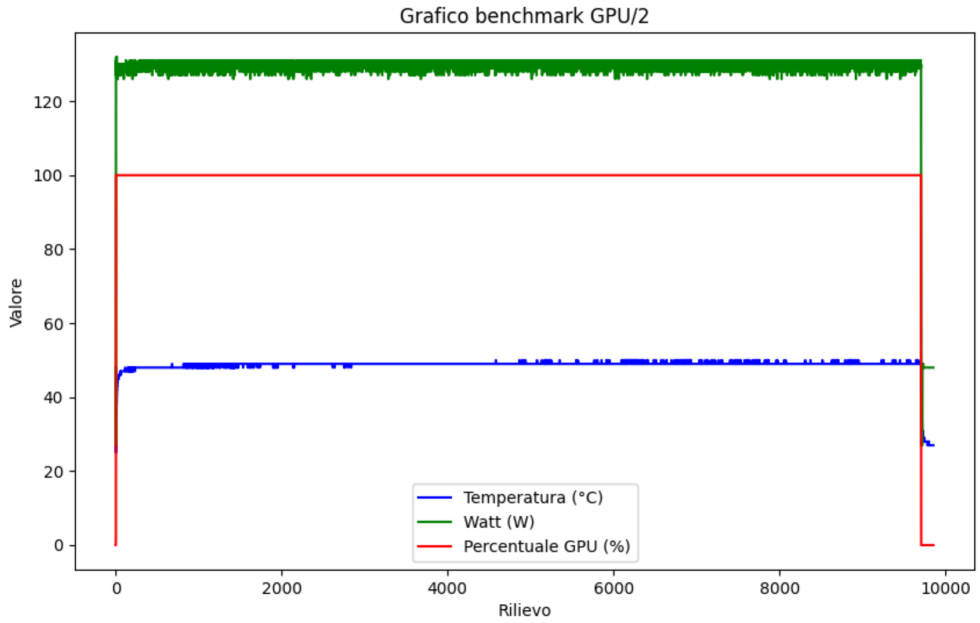

Fig. 4 — GPU/2: Card 2, A — ~48°C, ~128–130W. 12h, 100%, zero drift.

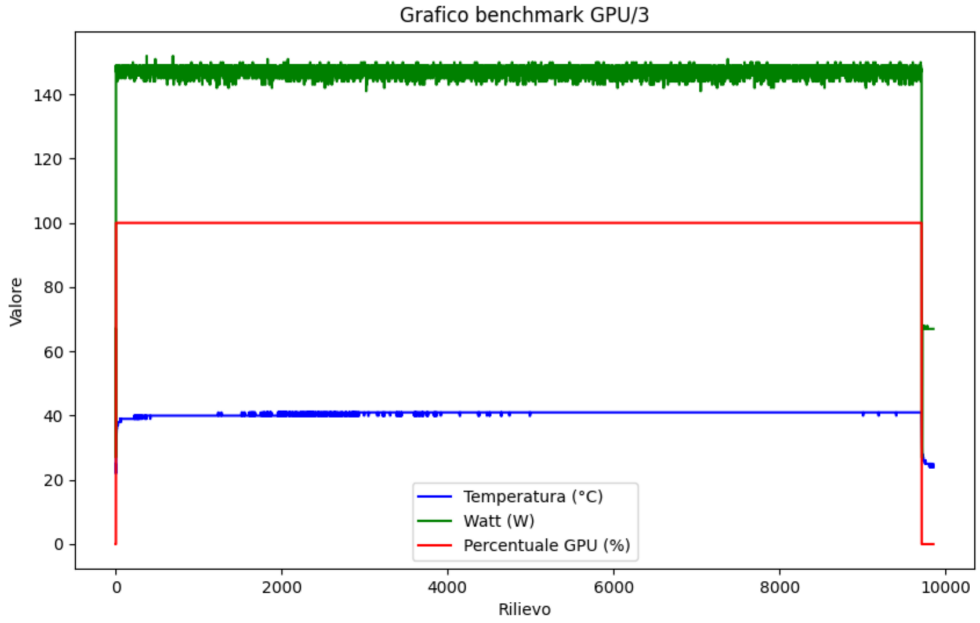

Fig. 5 — GPU/3: Card 2, B — ~40°C, ~145–148W. 12h, 100%, zero drift.

12-Hour Benchmark Data (continued)

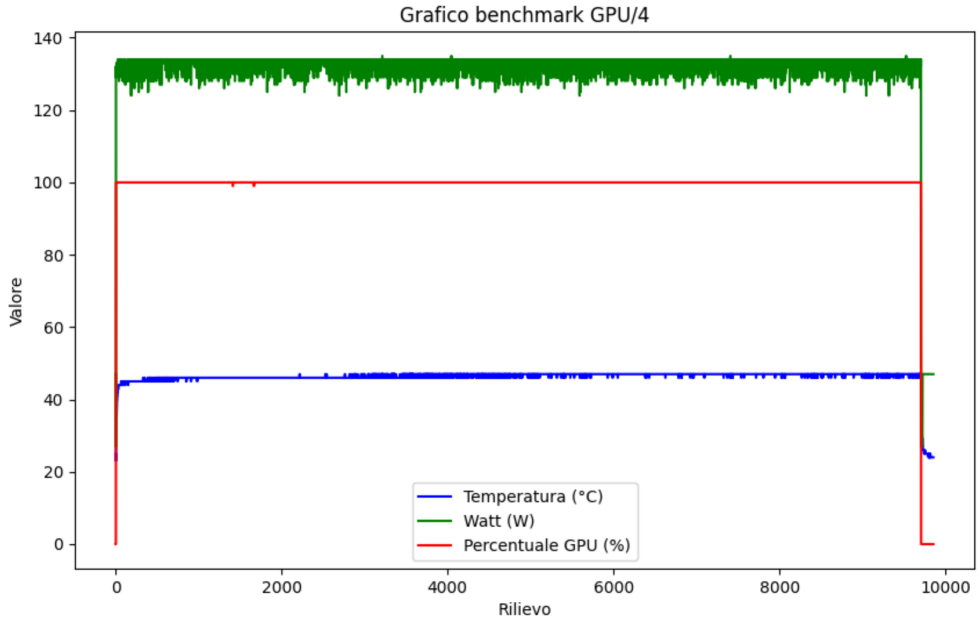

Fig. 6 — GPU/4: Card 3, A — ~45°C, ~128–133W. 12h, 100%, zero drift.

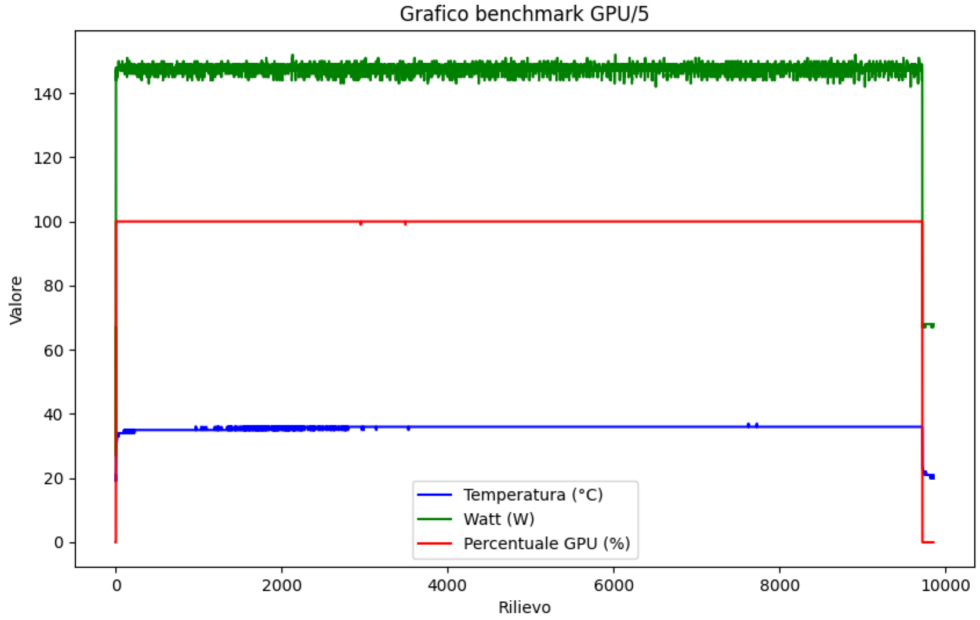

Fig. 7 — GPU/5: Card 3, B — ~35°C, ~145–148W (coolest). 12h, 100%, zero drift.

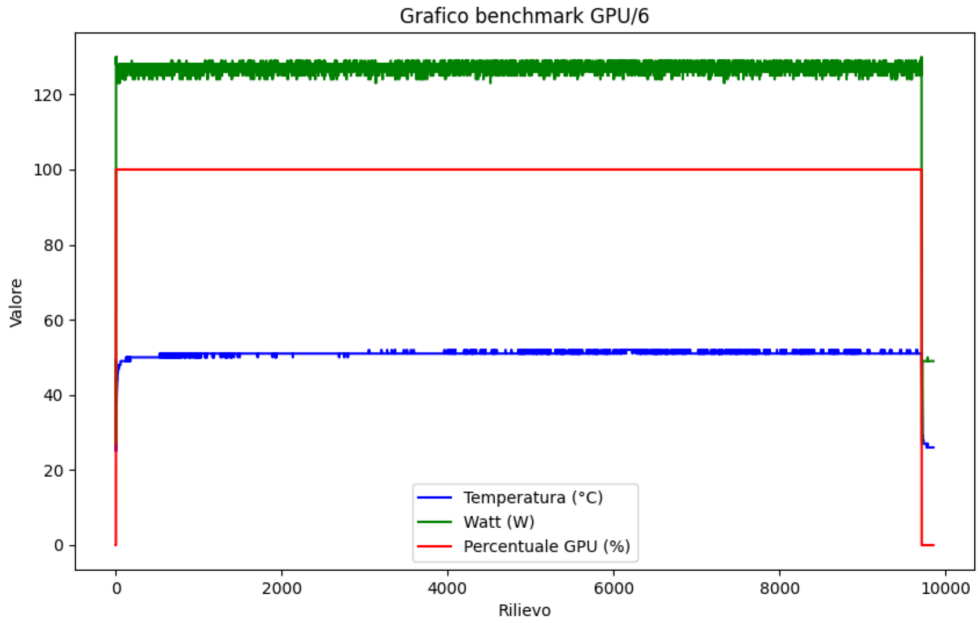

Fig. 8 — GPU/6: Card 4, A — ~50°C, ~125–128W (warmest). 12h, 100%, zero drift.

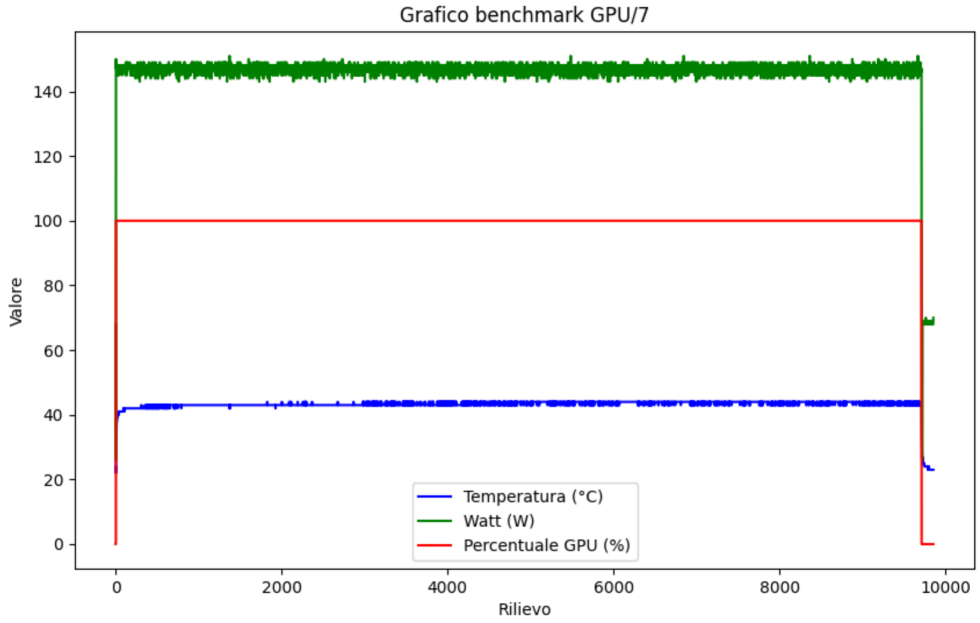

Fig. 9 — GPU/7: Card 4, B — ~42°C, ~145–148W. 12h, 100%, zero drift.

What the graphs demonstrate

The flatness of the blue temperature line is the critical feature. In a system with inadequate thermal design, temperature drifts upward as heat accumulates. The absence of any drift confirms the system reached thermal equilibrium within ~15 minutes and maintained it for ~11.5 hours. The sharp drop at sample ~9,800 (test termination) further confirms temperatures were load-driven, not ambient-driven.

Scaling

Once the single node was stable, I tested distributed configurations over InfiniBand — Mellanox adapters at 56 Gbps. A thermally and mechanically stable node makes every downstream problem solvable. An unstable node makes everything downstream pointless.

What this actually is

This project crossed five engineering domains: software, hardware architecture, thermodynamics, mechanical design, and cost engineering. Weakness in any single one would have made the system unviable.

What I built is not a server with GPUs. It is an integrated physical system where every component — from the GPU to the cable to the screw to the shape of the aluminum — participates in the behavior of the whole. Building it required learning disciplines I did not have at the start, making mistakes I could not have predicted, and iterating through five complete hardware revisions.

Most setups that claim to run AI locally work at startup. Very few work after 12 hours at full load. The difference between the two is not a software configuration. It is the physical engineering underneath — the structure, the airflow, the materials, the acoustics, the thermal envelope.

A system that works for 10 minutes is a demo. A system that works for 12 hours is infrastructure.

Benchmark data, thermal graphs, acoustic measurements, and iteration history are linked in the first comment.

Tommaso Bilotta — AI/HPC Engineer & DevSecOps Lead
linkedin.com/in/tommaso-bilotta-2158a812