IMPERIAL

Energy and Carbon Footprint of the x3d2 CFD Solver on AI-First and HPC-Class GPUs

Irufan Ahmed
Department of Aeronautics, Imperial College London

The question

Computational costs typically use execution time and ignores carbon emissions. We instrumented a CFD solver across three architectures to determine if standard carbon-accounting tools accurately predict physical telemetry.

The fastest machine is rarely the greenest and current carbon accounting tools does not capture this

Method

The CFD solver x3d2 was evaluated using a benchmark case across three grids. Platforms ran 3–10 randomised, `--exclusive` SLURM jobs per configuration with thermal cooldowns.

TGV exercises all numerical components against a known reference solution.

Table 1: Platform instrumentation.

Instrument	A100	GB10	GH200
NVML, 5 Hz	yes	yes	yes
RAPL via `perf`	yes	—	—
BMC whole-node	yes	—	—
ACPI module meter	—	—	yes
Wall-socket meter	—	yes	—

Sampling overhead $<1\,\%$; integration agreed to 0.04 %. Wall-time CV $\leq 1.0\,\%$ in all but one configuration. Total runs: 124 (GB10/A100), 36 (GH200).

How much of the machine NVML sees

Table 2: GPU-rail power as a share of system power, 256^3 double precision.

	GPU rail	system	share
GB10	29.7 W	102.1 W (socket)	29.1 %
A100	230.9 W	638.9 W (BMC)	36.1 %
GH200	376.7 W	460.6 W (module)	81.8 %

The GPU-rail is the only metric shared across all platforms, yet on GB10, it captures under a third of socket draw. below. A100 data aggregates both cards.

Offload first: no trade-offs within a machine

Table 3: 128^3, fully measured (RAPL + BMC), $n = 3$.

	CPU (24 core)	1× A100	ratio
wall time	7585.9 s	378.4 s	20.0×
whole-node energy	475.5 Wh	39.2 Wh	12.1×
nJ / cell-timestep	40 813	3 366	12.1×

GPU offloading yields time and energy gains. On GB10, this margin narrows (4.5× faster, 2.9× more energy-efficient at the socket) because the Arm CPU draws just 9.4 W above idle. Efficiency trade-offs only manifest between platforms.

During CPU runs, two idle A100s consumed 28 % of total node energy for zero work, nearly matching the CPU package.

A second GPU: the boundary dictates cost

Scaling to a second A100 yields just a 1.1× speedup (ideal: 2.0×). The resulting energy impact depends entirely upon the measurement boundary:

Table 4: One experiment, three boundaries, opposite signs.

boundary	1 GPU	2 GPU	ratio
working device	58.0 Wh	66.0 Wh	1.14×
both devices	82.2 Wh	87.9 Wh	1.07×
whole node	227.3 Wh	216.2 Wh	0.95×

Impact flips from a $+13.7\,\%$ penalty to a $-4.9\,\%$ saving. Poor scaling stems from hardware PCIe and cross-NUMA bottlenecks penalising FFT transposes.

Between platforms the trade-off is real

Bandwidth reduces time, not energy. Despite a 47× FP64 deficit, GB10 consumes 1.40× less GPU energy than A100 at 256^3, achieving the lowest energy per cell-step. GH200 finishes 1.80× faster than A100 at comparable energy per cell-step (617 vs 623 nJ).

Table 5: 256^3, double precision, one GPU. nJ/cell-step has idle power subtracted. Bandwidth is measured FP64 triad, not vendor peak.

	GB10	A100	GH200
wall time	7118 s	1281 s	710 s
GPU-rail energy	58.8 Wh	82.2 Wh	74.3 Wh
nJ / cell-step	548	623	617
bandwidth (GB/s)	230	1375	3532

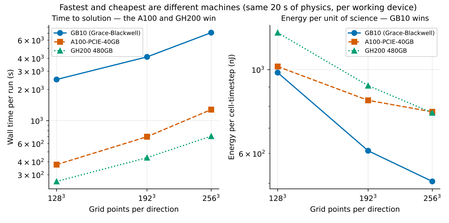


Figure 1: Resolution sweep: (a, b) wall time and (c, d) nJ/cell-step, in double and single precision. GH200 is always fastest. GB10 is the most expensive per cell-step at 128^3 and the cheapest by 256^3.

Single precision: bandwidth, not arithmetic

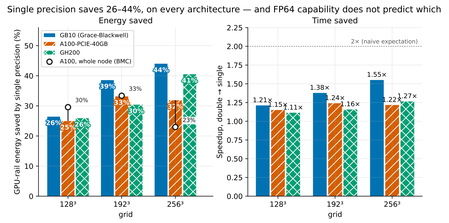


Figure 2: Single precision saves about 25 % of GPU-rail energy at 128^3 on every platform, diverging by 256^3 (GB10: 44.1 %, GH200: 40.4 %, A100: 35.8 %). Enstrophy stays within 0.01 % of double precision. Circles show whole-node A100 jobs: savings shift by up to six points and swap grid order.

Halving operand width halves memory traffic. GH200 approaches GB10 in energy savings (40.6 % vs 44.1 %) despite 157× more FP64. Energy reductions outpace time reductions by 8–20%, proving single precision also cuts power (9–25 %).

NVML agrees with the wall socket

Two identical units running identical software in the same environment exhibited a 7.4 % GPU power variance. A wall-socket meter confirmed this, matching NVML within 0.7 %.

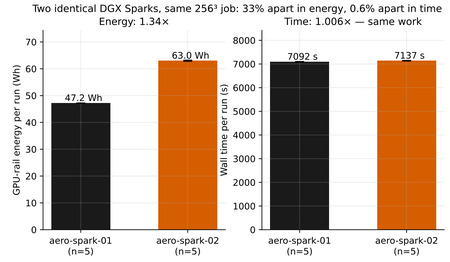


Figure 3: The same 256^3 job on two nominally identical DGX Sparks, both without a monitor attached.

Scheduling beats hardware

Grid carbon intensity varied 5.5× during the campaign. Across 12 months of London data, start time alone alters emissions of a single GB10 256^3 job by 28×. Conversely, the hardware energy gap between GB10 and A100 is merely 1.40×.

Could a tool have told us any of this?

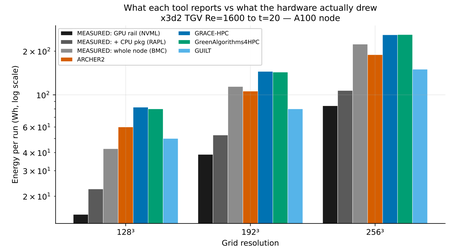


Figure 4: The four tools against all three measured layers on the A100 node, on jobs with correct SLURM GPU accounting. They span 0.64–1.11× of whole-node energy, but the layer you compare against decides the verdict.

On GB10, tools overestimated wall-socket energy by 1.9–2.9×. A100 predictions spanned 0.64–1.21× of node energy. Three tools assume 100 % utilisation (measured: 43 % A100, 61 % GB10). For carbon intensity, live-querying tools were accurate to 1 %, whilst static models averaged a 30 % overestimation.

The flaw is constant-based estimation: tools over-predict AI-first nodes by $2\times$.

Claude 5 (Anthropic) assisted drafting; the author refined all text. Measurements, analyses, and conclusions are strictly the author's own.