Multi-GPU Setups for Local AI: Worth It?

February 8, 2026 · by Mark Bartlett

Download this guide as PDF

> **Quick Answer:** Usually not. A single RTX 3090 ($800 used) handles every model up to 32B parameters. Multi-GPU only makes sense when you need more VRAM than one card provides — specifically, running 70B+ models that require 40-48GB. Dual 3090s cost $1,600-1,700 for the GPUs alone, plus $150-200 for a 1,200W PSU, and adding a second GPU to a model that already fits makes it 3-10% slower, not faster. If your target model fits on one card, buy one better card instead of two worse ones.

📚 **Related:** Multi-GPU Setup Guide · Used RTX 3090 Guide · VRAM Requirements · GPU Buying Guide

Everyone who runs AI locally eventually looks at their GPU and thinks: what if I had two?

The math seems obvious. Two RTX 3060s have 24GB of VRAM combined — same as a single 3090. Two 3090s give you 48GB — enough for 70B models. More GPUs, more VRAM, more capability. Simple.

Except it isn't. Multi-GPU setups have overhead, compatibility requirements, power demands, and cost structures that change the calculation entirely. Sometimes two GPUs are the right answer. More often, one better GPU gets you further for less money and less headache.

This guide is the decision framework. Not how to set up multi-GPU (we covered that in the setup guide) — whether you should.

The Promise vs The Reality

The pitch for multi-GPU is compelling: pool VRAM across cards, run models neither could handle alone, scale up by adding hardware instead of replacing it.

Here's what that looks like in practice:

Expectation	Reality
"Two GPUs = twice the speed"	Adding a second GPU to a model that fits on one makes it 3-10% **slower**

Expectation	Reality
“Two 3060s = one 3090”	Two 3060s have 24GB total VRAM but ~40% of the memory bandwidth, giving roughly half the tok/s
“I’ll save money with cheaper cards”	Dual 3060s ($400 total) + PSU upgrade ($100) = $500 for worse performance than a single 3090 ($800)
“Any two GPUs work together”	Mixed GPU sizes create bottlenecks — the fast card waits for the slow one
“Software handles it automatically”	Ollama auto-splits, but vLLM tensor parallelism requires matched VRAM sizes

The fundamental problem: GPUs in a multi-GPU setup don’t share memory. They each have their own VRAM, connected by PCIe — which is 20-60x slower than each GPU’s internal memory bandwidth. Every time data crosses that PCIe link, you pay a speed penalty.

Multi-GPU doesn’t give you a bigger GPU. It gives you two smaller GPUs trying to coordinate over a bottleneck.

When Multi-GPU Actually Makes Sense

There are four scenarios where adding a second GPU is the right call.

1. You Need 70B+ Models and Nothing Smaller Will Do

This is the primary use case. A 70B model at Q4 quantization needs ~40-45GB of VRAM. No single consumer GPU has that. Your options:

- **CPU offloading:** ~1 tok/s. Technically works. Practically unusable.
- **Dual 24GB GPUs:** 16-21 tok/s. Actually usable for chat and development.

If you’ve tested 32B models and they aren’t good enough for your use case — if you genuinely need 70B-class reasoning — dual 3090s are the most cost-effective path. Nothing else gets you there under $2,000.

The test: Before buying a second GPU, run Qwen 2.5 32B or Llama 3 32B on your single card. If the quality is sufficient, stop here. 70B is better, but the jump from 32B to 70B is smaller than the jump from 8B to 32B.

2. You Want Higher Quantization on Large Models

A 32B model at Q4 quantization fits on 24GB. The same model at Q8 — noticeably better quality — needs ~34GB. That's more than one card.

If you're doing work where output quality matters (creative writing, code generation, reasoning tasks), the quality difference between Q4 and Q8 on a 32B model can be worth the second GPU. You're not running a bigger model — you're running the same model better.

3. You're Serving Multiple Users

Multi-GPU scales nearly linearly for batch throughput. A single request doesn't benefit much beyond 2 GPUs, but 50 concurrent requests across 8 GPUs achieve ~800 tok/s total — each additional GPU adds real capacity.

If you're running a local AI server for a team, a household, or a small business, multi-GPU pays for itself in concurrent capacity. Each GPU adds KV cache space for more simultaneous conversations.

4. You Already Own Both Cards

If you've got a 3090 in your workstation and a 3060 in an old gaming rig, putting them in the same machine costs you nothing except a PSU upgrade. The 3060 won't match the 3090's speed, but it adds 12GB of VRAM for layers that would otherwise spill to CPU.

A 3090 + 3060 (36GB total) runs 70B models at Q3 — slower than dual 3090s, but dramatically faster than CPU offloading. Use what you have before buying what you don't.

When It Doesn't Make Sense

1. Your Model Already Fits on One Card

This is the most common mistake. Adding a GPU to run a model that fits on your existing card makes performance worse, not better.

Benchmarks with an 8B model on RTX 3090s:

GPUs	tok/s	Change
1	111.7	baseline

GPUs	tok/s	Change
2	108.1	-3.2%
4	104.9	-6.1%

Every GPU adds communication overhead. If the model fits on one card, that overhead is pure loss. No exceptions.

The rule: If your model fits in your GPU's VRAM, spend money on a faster single GPU instead of a second one. An RTX 4090 runs the same 32B model at 40-90% more tok/s than a 3090 — no coordination overhead, no PCIe bottleneck, no configuration needed.

2. You're Trying to Save Money with Two Cheap Cards

The "two cheap cards beat one expensive card" theory falls apart when you do the math:

Setup	Total VRAM	Cost	32B Q4 tok/s	Notes
2x RTX 3060 12GB	24GB	~$400 GPUs + $100 PSU	~18-22	PCIe bottleneck between cards
1x RTX 3090 24GB	24GB	~$800	~35-40	No overhead, full bandwidth
2x RTX 3090 24GB	48GB	~$1,700 + $150 PSU	~16-21 (70B Q4)	Only justified for 70B+ models

Same VRAM, but the single 3090 has nearly double the single-stream performance of dual 3060s. Why? Because the 3090's 936 GB/s memory bandwidth all goes to one model, while dual 3060s split 360 GB/s per card and add PCIe transfer overhead on top.

Two cheap cards give you more VRAM. They don't give you more speed per token. If both setups can run your model, the single better card always wins.

→ Check what fits your hardware with our Planning Tool.

3. You Haven't Considered the Total Cost

The GPU is not the only expense:

Component	Cost
Second GPU	$200-850 depending on card

Component	Cost
PSU upgrade (likely needed)	$100-200
NVLink bridge (if 3090s)	$80-120
Electricity (extra 200-350W, 24/7)	$175-300/year
Case with clearance for two 3-slot cards	$50-150 if your current one won't fit

A second RTX 3090 costs $800 for the card — but $1,000-1,200 when you factor in PSU, power cables, and the first year of electricity. That brings dual 3090s to $2,600-3,000 all-in for the first year.

For that money, you could buy a single RTX 4090 (used ~$1,500-2,200) with 24GB, no multi-GPU overhead, lower power draw, and better single-stream performance. Or you could rent cloud GPU time for the occasional 70B workload and keep your single-GPU setup for daily use.

The Budget Math

Here's the honest cost comparison for the most common decision points:

"I want to run 32B models"

Option	Cost	Performance	Verdict
Single RTX 3090 (24GB)	~$800	35-40 tok/s	**Best option**
Single RTX 4060 Ti (16GB)	~$450	Q4 only, tight fit	Budget option
Dual RTX 3060 (24GB total)	~$500	18-22 tok/s	Worse than single 3090

Winner: Single RTX 3090. Not close.

"I want to run 70B models"

Option	Cost (all-in, year 1)	Performance	Verdict
Dual RTX 3090 (48GB)	~$2,800	16-21 tok/s	**Most practical**
Single RTX 4090 + CPU offload	~$2,000	3-5 tok/s	Painful
Cloud API (occasional use)	~$50-200/mo	30-50+ tok/s	If you don't need 24/7 access

Winner: Depends on usage. Daily 70B use → dual 3090s. Occasional 70B use → cloud. No one should CPU-offload a 70B model and call it usable.

"I want the most VRAM possible under $2,000"

Option	Total VRAM	Cost
Dual RTX 3090	48GB	~$1,700
Single RTX 3090 + RTX 3060	36GB	~$1,000
4x RTX 3060	48GB	~$800 cards, but need workstation board + case

Four 3060s have 48GB for half the GPU cost — but need a workstation motherboard ($500+) with enough PCIe slots, a massive case, and a 1,500W PSU. The total system cost exceeds dual 3090s, and performance is significantly worse. Don't do this.

Software Support: What Actually Works

Not every tool supports multi-GPU, and not every tool supports it the same way.

Tool	Multi-GPU	Mixed Sizes	Parallelism	Notes
Ollama	Auto since v0.11.5	Yes	Pipeline only	Zero config, just works
llama.cpp	Yes	Yes (`--tensor-split`)	Both	Most control, best for mixed GPUs
vLLM	Yes	Pipeline only	Both	Tensor requires matched VRAM
ExLlamaV2	Yes	Yes (`--gs`)	Tensor (v0.3.2+)	Fast for EXL2 quantizations
Razer AIKit	Yes (wraps vLLM)	Via vLLM rules	Both	Turnkey Docker stack
Exo	Apple Silicon only	No	Layer sharding	Mac-only distributed inference

If you want zero configuration: Ollama. Install it, run your model, it splits automatically.

If you have mixed GPUs: llama.cpp. The `--tensor-split` flag lets you control exactly how work distributes.

If you're serving multiple users: vLLM with tensor parallelism (requires matched GPUs).

For detailed setup instructions, see the multi-GPU setup guide.

The Distributed Alternative

There's a third option between "one GPU" and "two GPUs in one machine": distributing inference across multiple machines on your network.

Instead of cramming two 3090s into one case with a 1,200W PSU, you keep each GPU in its own machine and coordinate over Ethernet. Your gaming PC runs the heavy layers. A mini PC handles embeddings. A laptop contributes spare cycles.

This is what projects like mycoSwarm and Exo are exploring. The advantage: no PSU upgrades, no motherboard lane splitting, no thermal problems from stacking cards. The disadvantage: network latency is slower than PCIe, so per-request speed is lower.

For workloads that can tolerate slightly higher latency — batch processing, async tasks, background summarization — distributed setups can use hardware you already own without any hardware modifications. Two machines with one GPU each might not beat two GPUs in one machine for raw speed, but they cost nothing extra if you already have the hardware.

It's early-stage technology. But for people who'd rather use what's in their closet than buy a second GPU and a bigger power supply, it's worth watching.

Decision Flowchart

Ask yourself these questions in order:

Does your target model fit on your current GPU? → Yes: Don't buy a second GPU. If you want more speed, buy a single faster GPU.

Is the model 70B+ parameters? → Yes: You need 40-48GB VRAM. Dual 24GB cards (3090s) are the practical answer. → No: It's probably a 32B model that needs Q8. Consider a single 4090 (24GB + faster) or dual 3090s if you also want 70B capability.

Do you already own a second GPU? → Yes: Put it in the machine. Free VRAM is free VRAM, even with overhead. → No: Calculate total cost including PSU, power, and cooling before deciding.

Are you serving multiple users? → Yes: Multi-GPU scales well for concurrent requests. Worth it. → No: You're optimizing for single-stream speed, where one better GPU always wins.

The Verdict

Multi-GPU is a solution to exactly one problem: running models that don't fit on a single card.

If your model fits on one GPU, a faster single card beats two slower ones every time. No overhead, no compatibility issues, no PSU upgrades, no configuration.

If your model doesn't fit — if you genuinely need 40-48GB for 70B models or high-quantization 32B models — dual RTX 3090s at $1,700 for the pair remain the most cost-effective path. Nothing else under $2,000 gets you 48GB of usable VRAM.

For everything in between, the answer is almost always: buy the best single GPU you can afford. An RTX 3090 at $800 used handles everything up to 32B parameters. That covers 90% of local AI use cases without ever thinking about multi-GPU.

The other 10% is where dual cards earn their keep. Just make sure you're actually in that 10% before spending the money.

📚 **Setup guide:** Multi-GPU Local AI: Run Models Across Multiple GPUs · llama.cpp vs Ollama vs vLLM

📚 **Hardware guides:** Used RTX 3090 Buying Guide · Best Used GPUs for AI · GPU Buying Guide · VRAM Requirements

Get notified when we publish new guides.

Subscribe — free, no spam

Source: https://insiderllm.com/guides/multi-gpu-worth-it/