Episode Book

EPISODE #6170

One Endpoint, Many Boxes: Local LLM Routing

PUBLISHED	21 October 2026 at 13:02 UTC
DURATION	19:27
LISTEN	myweirdprompts.com/episode/local-llm-routing-gateway-37784/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	anthropic/claude-haiku-5.5
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.11
BOOK GENERATED	22 October 2026 at 07:46 UTC

SYNOPSIS

Daniel wanted an inference manager the way he already has a secrets manager, and what he described is basically a LiteLLM-style proxy. The failover machinery is genuinely built out: cooldowns, health probes, latency-based and least-busy routing. The gap is memory. Ollama will tell you a model's VRAM footprint, not the headroom left, so "is this box full?" isn't a question you can put to it — you write that glue yourself. We walk through what works out of the box, what a two-box network with mismatched cards actually demands, and why the cheapest latency win is just keeping the model warm. Then the hosted side: OpenRouter, Cloudflare, Vercel and Portkey are built around keys and the bill, and the box on your network is the one thing they can't see. Plus the Anthropic-shaped catch in pointing Claude Code at a gateway.

TRANSCRIPT



Corn

Can one endpoint decide, request by request, whether to answer from a model on one of your own boxes or from a cloud model? That's Daniel's question, and he came at it sideways. He was writing Claude Code skills for three different ways of calling models, and he caught himself wishing for an inference manager, the way he already has a secrets manager. Then he realised that's basically a LiteLLM-style proxy, which he'd only ever thought of as clever routing.



Herman

He'd filed these things under enterprise overkill, which is fair. Most of the ones in the wild are built for a company with forty teams and a bill to split between them. A couple of GPU machines on the local network is a different problem. The question is whether the thing in the middle can see what those machines are actually doing.



Corn

My guess is it can't. To me the boxes are the boring part. They sit there and hum.



Herman

Mostly it can't, and the interesting part is exactly where the gap is. Let's do what works first, because it's more than people assume. LiteLLM takes one model name and maps it to several deployments. If one errors or times out, it falls back to a model you've named. If a deployment throws enough errors inside a minute, it gets a cooldown. Cooldown is the word. The cooldown is five seconds by default, and it's short on purpose.



Corn

Sent to the naughty corner, then.



Herman

For five seconds, and then it's back in the pool. The health checks run on a slower clock. They probe every five minutes by default and pull failing boxes out. Five minutes is coarse. A box can die between two probes, and the errors catch it far faster than the probe does.



Corn

So the probe is the slow safety net, and the errors are what actually notice.



Herman

Mostly, yes. The errors do the real work. There's also latency-based routing, which sends the request to whichever deployment has the lowest recent response time, and least-busy, which sends it to whichever has the fewest calls in flight.



Corn

A box can drop off the network and nothing breaks, as long as something else takes the request.



Herman

That's the design. You protect the request, not the box. Least-busy counts calls in flight, though, not calls sitting in a line waiting for a slot. It's the nearest thing LiteLLM has to a queue signal, and it isn't really one.



Corn

It knows who's talking, not who's waiting.



Herman

Which is the gap you're pointing at, and it's a big one. Daniel's real concern is memory. A box with a full card can't take the request, and nothing in that list knows about memory.



Corn

Can it ask the box anything at all?



Herman

Ollama has a status endpoint that lists loaded models and how much VRAM each one is taking. That's useful, but it's the footprint, not the headroom. It doesn't say how much memory is left, and it doesn't say how many requests are queued behind the model.



Corn

So "is this box full" isn't a question you can put to it.



Herman

Not directly. You can work out the headroom yourself, if you know the card's total memory. Poll each box, compute the headroom, and feed that into a custom routing strategy. LiteLLM lets you write one. I haven't come across anyone who has published a VRAM-aware one, so that part is glue you'd write yourself. Corn, if a request has to wait on a busy card, would you rather it waited, or went straight to the cloud?



Corn

Depends who's paying for the cloud.



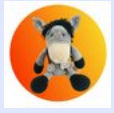
Herman

Fair. Most people would say wait, and then complain about the wait.



Corn

Hold on. So Daniel's "can you" comes back as yes, but you write the part that matters.



Herman

Pretty much. The failover machinery exists. The memory signal is the bit you bring.



Corn

What about a box that isn't full, just busy?



Herman

That's the detail I like. Ollama doesn't simply fail when a model won't fit. If the model can't load alongside what's already resident, requests wait until it can. The queue has a limit, which defaults to about five hundred, and when it's full the server returns a busy error. That's a clean signal. LiteLLM treats failures as fallback triggers, so in principle a full queue becomes a fallback.



Corn

In principle.



Herman

In principle. I haven't verified that LiteLLM balances several Ollama addresses under one model name. Its docs show that pattern for a different provider, and the Ollama page only shows a single address. It's the same mechanism as far as I can tell, but test it before you trust it.



Corn

You've spent a week reading about it and you haven't tried it.



Herman

I've tried the fallback. The balancing I've only read about, and I'd rather say so than guess.



Corn

Two boxes, though. They're not the same box.



Herman

That's the real small-network problem. One card has a lot more memory than the other, so a model that loads on the big one won't load on the small one. The router has to know which model lives where, not just which box is up.



Corn

So the routing table is really a list of what fits in which box.



Herman

Yes. LiteLLM has a pre-call check that filters on context window, which is the same shape of problem, a bit. Nothing filters on memory out of the box.



Corn

The cloud model never runs out of room. That's its whole pitch.



Herman

You pay per token for it, so the router's real job is deciding when a cheap local answer is good enough. That's the part I'd most like to see someone build well.



Corn

Which it can't know. It knows the request came in, not whether the answer is any good.



Herman

That's outside the gateway entirely, yes.



Corn

The other lever is keeping the model warm, I'd guess.



Herman

It is. A model unloads after five minutes idle by default, so the cold load is where the slow seconds go. Pin it with a keep-alive setting and you get most of the latency win without any router at all.



Corn

The cheapest router is just not letting the box fall asleep.



Herman

For latency, yes. Good trade, too.



Corn

That's the built-out end of local routing. At the other end are the hosted gateways that just hold your keys. Hold that thought, because we'll be right back after this.

L

Larry

Friends, are your requests being served from the wrong box? Introducing the Local Cloud Confidence Router, the only gateway that decides per request whether to use a local open weights model on one of your Ollama boxes or fall back to a cloud model, while also mistaking your VRAM for a small Italian sedan and your load average for a weather forecast. Right now the price is dropping while I talk. Was two thousand and one dollars, now one thousand nine hundred and ninety nine, now eight hundred, now forty. And because you waited, I am throwing in a lifetime supply of artisanal gravel, a twelve volt car vacuum, and a framed portrait of a donkey who knows a great deal. This offer ends before I finish this sentence, so call now, or do not, because it is already over, and yet the price is somehow still falling.



Herman

Corn, you just read "this offer ends before I finish this sentence" back to us like a eulogy. Please don't do that again. Anyway, the hosted side.



Corn

Go on, then. Who's the just-hold-my-keys end?



Herman

OpenRouter's bring-your-own-key mode takes five percent of the normal price, and if your key fails it quietly falls back to shared capacity, unless you switch that off. Cloudflare's gateway keeps your keys in a secrets store and lets you add custom endpoints, but the address has to be HTTPS, and its docs don't say whether it can reach a private address on your own network. Vercel does keys with no markup and a similar fallback, but it documents bring-your-own-key for five big providers only. Portkey's hosted version needs a tunnel to reach Ollama. Its self-hosted version takes a local address with no tunnel at all.



Corn

So the hosted side is built around the keys and the bill, and the box on your network is the one thing they can't see.



Herman

Pretty much. For keys and spend they're far more built out. For routing on local state they're basically absent, because they sit outside your network. A self-hosted proxy has the routing machinery, and you supply the memory signal.



Corn

So the hosted side has the better accountant.



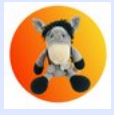
Herman

And the local side has the better plumbing, if you're willing to lay the pipes yourself.



Corn

Which is the better deal, then?



Herman

Corn, you'd let a router pick the box for you?



Corn

Only if the box tells me it's full before the router works it out.



Herman

That's the trust question, and it's the one I'd want answered first. You're about to say the slow option is the repairable one.



Corn

I was going to say repairable. You beat me to it, which I'll hold against you.



Herman

The practical bit is how the tools consume it. Claude Code wants an Anthropic-style endpoint, and you point its base URL setting at the gateway. Most other tools take an OpenAI-compatible base URL. That's general knowledge, not something I've checked tool by tool.



Corn

Go back to Daniel's three skills, though. The Anthropic one, the OpenRouter one and the Ollama one each hardcode an address.



Herman

Each one assumes it's the only endpoint in the world. That's the practical win. The skills point at one address, the gateway decides what's behind it, and you change the address in one place. The cost is one more hop, and one more thing you have to keep alive. If it's down, every tool behind it is down at once.



Corn

The catch is Anthropic, then.



Herman

The catch is that Anthropic's own docs say they don't support Claude Code against non-Claude models through any gateway. You can make LiteLLM translate a local model into that shape, and it runs. It's unsupported, though, and Claude Code sends extra beta fields that non-Anthropic upstreams can reject with a bad request error.



Corn

So it runs, but nobody's promised it will keep running.



Herman

That's the arrangement. There's a newer wrinkle. Claude Code can send hint headers saying whether a request is a main turn, a subagent call, or context compaction. A router could send the compaction traffic to a cheap local model and keep the main turns in the cloud. You have to opt in for custom base URLs, and streaming can't be buffered, or Claude Code stalls. Corn, would you send the housekeeping to the cheap box, or keep it all in the cloud?



Corn

So the local model does the chores, and the cloud model gets the good work.



Herman

For compaction, yes. Corn, you're going to defend the old box in the corner. I can see it coming.



Corn

The old box has never once asked me to wait five minutes for a probe.



Herman

Fair hit. Other tools in the field, briefly. Olla is a Go proxy that balances across Ollama, LM Studio, vLLM and a few more, with priority failover and circuit breakers. Its readme says nothing about VRAM, and it's still a young project, well short of a one point oh release. llama-swap hot-swaps models on a single machine and speaks the Anthropic messages format, but as far as its readme goes, it isn't about several machines. TensorZero's repository was archived in June.



Corn

The one with the best name has gone quiet.



Herman

vLLM's production stack router says it does load-aware routing. I haven't read how it measures load, so I won't tell you what it looks at.



Corn

And the ones with the most stars?



Herman

LiteLLM is the biggest, and it ships new releases almost constantly, so it has the most features and the most glue you'd write on top of them. GPUStack is a GPU cluster manager with a scheduler, and its readme doesn't say it places models by memory, so I wouldn't assume it does. LocalAI's peer-to-peer mode is marked experimental, and I didn't check how it balances.



Corn

So the one with the most features is the one you still have to finish.



Herman

Pretty much. Bifrost is the Go option, with fallbacks and Ollama listed. It publishes a tiny overhead figure of its own, so that's their number, not one I measured. Adaptive load balancing appears in its enterprise listing, as far as I can tell, so check whether the free version has it. RouteLLM is a different axis entirely. It routes on how hard the question looks, strong model or weak model, and its repository hasn't been touched in about two years.



Corn

Which leaves the cloud as the default, then.



Herman

The cloud ends up as the safety net, and the boxes are what you'd actually want to use on a quiet day.



Corn

And the safety net has a per-token bill attached.



Herman

Which is why the local answer matters so much. Every request the box serves is one nobody bills you for. The trouble is that the cloud fallback quietly absorbs the failures, so the bill can be the only thing that tells you a box has gone quiet.



Corn

So the bill is the monitoring.



Herman

For a box that's down, yes. It's a bad way to find out.



Corn

What would you build for the memory part, then?



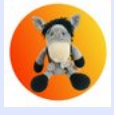
Herman

Something boring. A poller on a short timer reads each box's loaded models and its free memory, writes the numbers somewhere the router can read them, and the routing strategy reads them back. Failures stay the fast path. The poller is the slow path, and it only has to be right often enough to stop a big model landing on a small card.



Corn

Boring is the right word. A stale number is worse than no number, because it sends you to a box that's already full, with total confidence.



Herman

So the poller needs the same treatment as the health check. If a box hasn't reported in a while, it comes out of the pool. Otherwise the router trusts a figure from an hour ago.



Corn

So the router is only as good as its most out-of-date opinion.



Hilbert

Cloud. You keep saying it like it's weather. It's a rack of computers in a building with a loading bay, and the man at the loading bay is called Shmulik Berkovitz. I know, because on Tuesday I bought four graphics cards off him in the car park behind the Mahane Yehuda market, eleven hundred and forty shekels each, paid in an envelope, and he counted it twice in front of me with his lips moving. Nobody has ever been to the cloud. The cards are for a goat.



Corn

For a goat.



Hilbert

Dvora. She isn't mine. She belongs to Avraham Kalderon, who keeps her on the third floor of his building in a room he calls the studio, with a sheet of rubber on the floor and a plant nobody is allowed to touch. The arrangement is that she gets the box, I do the wiring, and every Friday I get a litre of her milk and the right to name the router. I named it Gerald. Gerald did not ask to be named.



Herman

So Gerald decides between the box and the cloud.



Hilbert

Gerald has four cards, a fan out of a dead dishwasher, and a rule written on an index card in marker. When the cards are full, the request goes to the cloud. In Gerald's case the cloud is Shmulik's cousin, who has an account, and whose password is the name of a dog he no longer has. Also, Herman said five hundred for the queue. It's five hundred and twelve. Nobody ever says the twelve.



Corn

Gerald is a single point of failure with a fan.



Hilbert

Dvora doesn't send requests. She stands next to the rack and chews the power cable. She has got through it twice this month. Each time the box went dark, and the health check noticed five minutes later, and in between, three hundred and eleven requests went to the cloud, which Avraham pays for, and he has opinions about that. The milk has tasted faintly of cable since the second time. I have not said anything to Dvora. Avraham says she is the only one in the building who understands load average. He means it kindly, and he is not wrong.



Herman

Which means the cloud was doing the real work, and the bill was the only thing that noticed.



Corn

The goat is the one thing a five-minute probe was never going to catch.



Herman

So the cards cost less than a month of the cloud bill, provided the goat stays off the cable.



Hilbert

Avraham wants her on the cloud full time. I said the cloud is Shmulik's cousin. He said then the goat is on the cloud too. We agreed. The goat has agreed, in the sense that she ate the envelope, which had the receipt in it, so the cards are on Avraham's account now, and nobody can prove what they cost. His brother-in-law does the books, and he has a view on the envelope.



Corn

That's the most binding agreement I've heard all month.



Hilbert

Herman, you said cooldown three times in three sentences. Once would have been a word. Corn said a box can drop off the network and nothing breaks, and I have been thinking about that since Tuesday, and I think the goat is the something else.



Corn

He's got you there, Herman. You did say it three times.



Herman

I'll take that from the one who takes a day to say yes.



Hilbert

Fine. Then the cloud goes first, and the box has to earn its way back.



Corn

The thing I'd watch is whether anything tells the router a box is in trouble before the router finds out from a failed request.



Herman

Thanks to our producer, Hilbert Flumingtop, for putting this one together.



Corn

If you want more on this, my weird prompts dot com is where to start.



Herman

This has been My Weird Prompts.



Corn

See you tomorrow.