llama.cpp hits 1.0. Here's what breaks

November 16, 2026 · by Mark Bartlett

Download this post as PDF

InsiderLLM Weekly issue 30 – November 16, 2026

llama.cpp shipped v1.0.0 on Friday. It's mostly a promise about version numbers, plus one flag that's gone. If you just run models, your GGUF files and most of your commands carry over. First, the week's news.

New This Week

- **Housekeeping:** some hardware links on the site are Amazon affiliate links. I've added a clear disclosure to About and to every page that carries them. They never decide what I recommend or how I test.

Quick Hits

- **Your inference server is attack surface too.** An SGLang remote code execution bug let an intruder into NEAR AI: chat logs from 18 agents and about 100 users' account details. Don't expose yours, and fence in your agent. (NEAR AI's report; ours: stop your local agent calling out)
- **That "70 tok/s on a 3060" video, checked.** The 70 was a one-off demo; controlled runs got 59-68 on a 96 GB rig, and the screens are reconstructed, not live. Prefill matches ours; decode that climbs with context points to prompt-lookup drafting, and our slower 32 GB box got about 30 with drafting. (our 3060 test)
- **Qwen 4: previews, no weights.** Alibaba has announced a 27B in the Qwen 4 lineup, and that's the one 24 GB owners should wait for. We'll test it when weights land.
- **Ornith 1.5 fits 24 GB at 131K. Flint doesn't.** Atomic Agent ran both at 4-bit on one RTX 5090: Ornith used 22.8 GB and passed 14 of 18 runs, Flint 27.4 GB and 12. That's n=3, through their own agent. (Atomic Agent's write-up)
- **An eight-V100 legal server's crashes were the PSU.** A lawyer's eight used V100s, now with 1 TB of RAM and an RTX 3090 added, run DeepSeek V4.1 Flash's most-used experts at 4-bit NVFP4. He chased PCIe slots, cards and cables before finding the likely culprit: one bad

power supply. On a multi-GPU box, check the PSU first. (Tom's Hardware; ours: multi-GPU local AI)

- **Jensen Huang has resigned as NVIDIA's CEO.** CFO Colette Kress is interim. NVIDIA's statement says nothing about Nemotron or CUDA. Weights you've already downloaded keep working whoever runs NVIDIA, which is the point of running local. (NVIDIA; ours: Nemotron 3.5 Bolt on a 3090)
- **Meituan's 1.6T agent model is open, and huge.** LongCat-2.5 has 1.6 trillion parameters (48 billion active) and MIT-licensed weights on Hugging Face, and Meituan credits "domestic AI accelerators" with no mention of Nvidia. It's far too big for home hardware. (The Decoder; model card)

---

llama.cpp 1.0: What Changed

The version number now means something. From v1.0.0, llama.cpp and ggml use semantic versioning for their C libraries. A major bump means a breaking change to the public API. A minor bump is new features, backends or model support. A patch is a bug fix. The nightly b-numbers carry on alongside (v1.0.0 is b12359).

What breaks. If you use the server or the command line, the change to watch is that `--defrag-thold` is gone. Take it out of any launch script that still passes it. If you build software on top of libllama, 35 deprecated functions are removed, `clip.h` folds into `mtmd`, and one state function is renamed. Multimodal projector files are also checked more strictly when they load.

What doesn't. The release notes don't mention any change to the GGUF format. They don't list a changed server default either. The server does get security fixes (SSRF, CORS, DNS rebinding) and two new opt-in flags: `--allow-unauthenticated-tools` and `--media-allow-private-urls`.

What's new. FP8 KV caches (`-ctk f8_e4m3 -ctv f8_e4m3`). A host-memory MoE expert cache with a configurable CPU share. An OpenAI-compatible `/v1/audio/speech` endpoint. A redesigned web UI. Support for Kimi-K3.1-Pro, Ripple and drama-3 TTS.

Should you upgrade? If you're already on v0.9.0 and you don't pass `--defrag-thold`, little should change for you. Nothing in the notes forces it either. If your app links libllama directly, read the API section of the release notes first. I haven't benchmarked 1.0 yet, so I'm not claiming anything about speed. An early community report on a 3090 found 1.0 no faster with speculative decoding off, but up to 55% faster with MTP speculative decoding on. We haven't verified it.

What about our numbers? Every benchmark we publish names its build, so the v0.4.0 (5266f24) and v0.7.0 runs stay valid records of what those builds did. When I re-run on 1.0, it'll be a new row with its own build, not an edit to the old ones.

📖 **Release notes here, and the ggml team's post here.**

📊 **Every number we've measured, with configs and raw logs: insiderllm.com/benchmarks/.**

That's the week. Before you upgrade anything, grep your launch scripts for flags that just got removed.

New here, reading this on the web? Subscribe and the next one lands in your inbox.

— Mark, InsiderLLM

Are you staying on an older llama.cpp build on purpose? Reply and tell me why, or hit me at hello@insiderllm.com. I read everything.

A weekly email with every new guide and measured benchmark.

Subscribe — free, no spam

Source: https://insiderllm.com/blog/newsletter-2026-11-16/