SÉCURITÉ & IA

Une IA locale qui tient la route

Matériel, modèles MoE, llama.cpp, Colibri, LoRA : ce que demande vraiment une IA dans ses murs, avec nos mesures, et pourquoi une base documentaire vérifiable compte plus que la carte graphique.

Publié le 26 novembre 2026 — Imprimé le 27 novembre 2026



Eine KI im eigenen Haus, damit die Dokumente das Unternehmen nicht verlassen: Die Idee überzeugt, und die Angebote häufen sich. Bevor man von Grafikkarten spricht, muss man wissen, was man sie tun lassen will und woran man erkennt, dass sie nicht irgendetwas erzählt. Hier ist, was wir gelernt haben, als wir sie bei uns betrieben haben.

Zuerst der Speicher, dann die Leistung

Ein Sprachmodell muss vollständig in den schnellen Speicher passen, damit es rasch antwortet. Die Menge des Speichers und dann seine Geschwindigkeit entscheiden darüber, was eine Maschine leisten kann. Drei Familien teilen sich das Feld:

- **Die RTX-Grafikkarten.** Sehr schnell, aber ihr Speicher ist begrenzt: 12 bis 32 GB bei den gängigen Karten, 96 GB bei der RTX PRO 6000 Blackwell, deren Preis sich mit der Speicherknappheit in wenigen Monaten fast verdoppelt hat. Ideal für ein kleines oder mittleres Modell, das viele Personen bedient.
- **Die Maschinen mit Unified Memory**: AMD-Strix-Halo-Mini-PCs, NVIDIA DGX Spark, Macs mit Apple-Chip. Prozessor und Grafikkarte teilen sich bis zu 128 GB, wodurch sich grosse Modelle in einem kleinen, leisen Gehäuse laden lassen. Ihr Speicher ist langsamer, sodass jede Maschine weniger Personen gleichzeitig bedient.
- **Die Server**, wenn Dutzende von Benutzern gleichzeitig bedient werden müssen. In einem KMU selten nötig.

Die Preise sind 2026 in die Höhe geschnellt. Ein Strix-Halo-Mini-PC mit 128 GB, der im Frühling zwischen 1’500 und 2’200 US-Dollar kostete, kostet heute 3’500 bis 4’300. Man muss also am Tag des Kaufs kal-

kulieren und mehrere mittlere Maschinen einer einzigen sehr teuren vorziehen: Wenn eine ausfällt, laufen die anderen weiter.

MoE-Modelle verändern die Lage

Die Modelle mit «Mixture of Experts» (MoE) aktivieren für jedes erzeugte Wort nur einen kleinen Teil ihrer Parameter. Ein Modell mit 80 oder 125 Milliarden Parametern beansprucht jeweils nur einige Milliarden davon: Es belegt viel Speicher, antwortet aber fast so schnell wie ein kleines Modell.

Genau das passt zu den Maschinen mit Unified Memory. Umgekehrt wird ein grosses «dichtes» Modell, das bei jedem Wort alle seine Parameter einsetzt, dort sehr langsam.

Was wir gemessen haben



Unsere Arbeitsmaschine ist ein Strix-Halo-Mini-PC mit 128 GB Unified Memory, unter Fedora, mit llama.cpp als Engine. Messungen von Anfang November 2026, im realen Einsatz durch Entwicklungsagenten:

- **Qwen3-Coder-Next**, ein auf Code spezialisiertes MoE-Modell: 46 bis 47 erzeugte Tokens pro Sekunde, 750 bis 875 gelesene pro Sekunde, 49 GB Speicher. Über einen Tag: 773 Anfragen und 2.4 Millionen gelesene Tokens, ohne Absturz.
- **Qwen3.8-Flash-Next**, ein grösseres Allzweckmodell: 25 Tokens pro Sekunde, etwa 500 beim Lesen, 80 GB Speicher.

Diese Zahlen gelten für eine einzelne Maschine. In einem Cluster bedient jedes Gehäuse seine eigenen Anfragen: Die Durchsätze addieren sich, und vier bis sechs Maschinen leisten grundsätzlich vier- bis sechsmal so viel.

Zwei praktische Lehren. Die Kontextlänge hat eine physische Grenze: Über 131’072 Tokens hinaus lief der Speicher über, und die Maschine musste neu starten. Und eine Maschine kann nicht alles gleichzeitig: Ein Modell bereitzustellen, Agenten laufen zu lassen und parallel Bilder zu berechnen, hat sie überlastet. Man verteilt oder man plant.

llama.cpp, Colibri und die anderen

llama.cpp ist die Referenz-Engine, um ein Modell lokal zu betreiben. Es läuft auf fast allem (NVIDIA- und AMD-Karten, Unified Memory, Mac), bietet eine Schnittstelle, die mit den gängigen Werkzeugen kompatibel ist, und lässt sich fein einstellen. Das ist die, die wir verwenden.

Colibri ist eine neuere Engine, die die «Experten» eines MoE-Modells zwischen Grafikspeicher, Arbeitsspeicher und Festplatte zirkulieren lässt. Sie verspricht, Modelle, die für den verfügbaren Speicher viel zu gross sind, auf gewöhnlicher Hardware laufen zu lassen. Sie bezeichnet sich selbst als experimentell: Wir evaluieren sie, empfehlen sie aber noch nicht für den Produktivbetrieb.

Ein Modell pro Bedarf

Es gibt kein einzelnes Modell, das alles gut kann. Was funktioniert, ist ein kleiner Katalog:

- **Redaktion und laufende Dokumente**: ein mittleres, schnelles Allzweckmodell.
- **Code**: ein spezialisiertes Modell wie Qwen3-Coder-Next, für kurze und begrenzte Aufgaben.
- **Bilder und Bildschirme**: ein Vision-Modell, um einen Screenshot, eine gescannte Rechnung, eine Oberfläche zu lesen.
- **Kontrolle**: ein zweites Modell, das das erste bei dem gegenliest, was zählt: Zahlen, Daten, Beträge.

Für Komplexes und Nichtvertrauliches stehen die grossen Online-Modelle weiterhin zur Verfügung. Entscheidend ist, für jede Aufgabe zu wissen, wohin die Daten gelangen.

Eine echte Basis statt Tokens

Viele Angebote lassen das Modell bei jeder Frage Hunderte von Seiten neu lesen. Das ist speicherintensiv, langsam, sobald mehrere Personen arbeiten, und nicht überprüfbar: Das Modell kann vermischen, verzerren oder erfinden.

ClearMyDocs v4, unsere Dokumenten-Engine, geht vom anderen Ende aus. Die Dokumente werden einmal verarbeitet, und jeder übernommene Fakt wird Wort für Wort zitiert, mit dem Dokument, aus dem er stammt. Eine automatische Kontrolle weist alles zurück, was sich nicht auf eine Quelle stützt. Das Modell schreibt anschliessend auf der Grundlage dieser geprüften Fakten, und jede Aussage des Ergebnisses bleibt mit ihrem Dokument verknüpft.

ClearMyDocs v4 läuft seit mehreren Monaten mit einem lokalen Modell vernünftiger Grösse, an echten Rechts- und Fakturierungsdossiers. Das ist die Antwort, die wir auf die Frage geben, auf die es ankommt: Was tut man, wenn sich das Modell irrt? Man glaubt ihm nicht aufs Wort: Man prüft, automatisch, gegen die Quelle.

Und das Training: LoRA oder QLoRA?

LoRA erlaubt es, ein bestehendes Modell an ein Fachgebiet anzupassen, indem nur eine kleine zusätzliche Schicht trainiert wird, ohne das ursprüngliche Modell anzutasten. QLoRA tut dasselbe an einem komprimierten Modell, was mit viel weniger Speicher auskommt, auf Kosten von etwas Genauigkeit.

Wir experimentieren damit, um einem lokalen Modell die Regeln eines Projekts anhand von bereits als gut beurteilten Arbeiten beizubringen. Für die meisten Unternehmen ist das aber nicht der erste Schritt. Eine gute Dokumentenbasis und Kontrollen bringen mehr und schneller als ein neu trainiertes Modell. Die Anpassung kommt danach, wenn man genau weiss, was fehlt.

Zusammengefasst

- Den Speicher vor der Leistung wählen und mehrere Maschinen statt einer einzigen.
- Auf Maschinen mit Unified Memory MoE-Modelle bevorzugen.
- Ein Modell pro Bedarf und ein zweites, das das erste kontrolliert.
- Eine überprüfbare Dokumentenbasis statt Seiten, die bei jeder Frage neu gelesen werden.
- An den eigenen Dokumenten messen, bevor man entscheidet.

ClearMyDocs und unsere Software · Sprechen wir über Ihr Projekt