r/LocalLLaMA • 2d ago
Juan_Valadez

Qwen3 Coder Next on 8GB VRAM

Tutorial | Guide

Hi!

I have a PC with 64 GB of RAM and an RTX 3060 12 GB, and I'm running Qwen3 Coder Next in MXFP4 with 131,072 context tokens.

I get a sustained speed of around 23 t/s throughout the entire conversation.

I mainly use it for front-end and back-end web development, and it works perfectly.

I've stopped paying for my Claude Max plan ($100 USD per month) to use only Claude Code with the following configuration:

```
set GGML_CUDA_GRAPH_OPT=1
```

```
llama-server -m ../GGUF/qwen3-coder-next-mxfp4.gguf -ngl 999 -sm none -mg 0 -t 12 -fa on -cmoe -c 131072 -b 512 -ub 512 -np 1 --jinja --temp 1.0 --top-p 0.95 --top-k 40 --min-p 0.01 --repeat-penalty 1.0 --host 0.0.0.0 --port 8080
```

I promise you it works fast enough and with incredible quality to work with complete SaaS applications (I know how to program, obviously, but I'm delegating practically everything to AI).

If you have at least 64 GB of RAM and 8 GB of VRAM, I recommend giving it a try; you won't regret it.

152 65 Share

MezmoData • Promoted

Tired of every incident turning into a 3-hour team huddle? Mezmo's AI SRE agent finds the "why" instantly, cutting the guesswork and giving you back your night.

Learn More mezmo.com

Join the conversation

Sort by: Best Search Comments

iamapizza • 2d ago

roosterfareye • 2d ago

I upgraded to 64gb of ddr4 around 3 years ago. It was overkill, at the time, but dumb luck paid off!

15 Reply

mindwip • 2d ago

Lol this is me too

1 Reply

SkyFeistyLlama8 • 2d ago

I regret getting 64 GB on a laptop. Qwen Next Coder 80B at Q4 takes up around 50 GB RAM when running so I don't have much memory left over for other programs.

I'm getting around 10 t/s on ARM64 CPU inference. Given the quality of the replies, it's more than fast enough. If I need more free RAM, then I go one step down to Qwen Coder 30B for function level work. Qwen Next is good enough to work with multiple modules and smaller entire codebases.

2 Reply

No_Swimming6548 • 2d ago

GLM 4.7 FLASH isn't bad at all 😮‍💨

1 Reply

JoeyJoeC • 2d ago

Everyone laughed at me for getting 64gb.

1 Reply

000loki • 2d ago

Do you have ddr4 or 5?

2 Reply

UnknownLegacy • 2d ago • Edited 1d ago

I have a similar system and I just cannot break 17 t/s.

Ryzen 7 5800X3D
64 GB ram
RTX 5080 16GB

I'm quite new at this, so I kind of took a combination of what everyone said in this thread here. I tested a bunch of different arguments and speed ran them with a fizzbuzz generation test. This one was the fastest (not by much though, 17 vs 16.5 t/s).

.\llama-server --model models\Qwen3-Coder-Next-MXFP4_MOE.gguf --temp 1.0 --top-p 0.95 --min-p

This is only using 32GB of my system ram (with Windows taking 16GB itself...). I feel like I'm missing something...

EDIT: I believe I found the issue. CUDA 13 vs CUDA 12 build of llama-server. I was using CUDA 12 build when I had CUDA 13 installed.

.\llama-server --model models\Qwen3-Coder-Next-MXFP4_MOE.gguf -c 65536 -fa 1 -np 1 --no-mmap --host 0.0.0.0 --port 8080 --temp 1.0 --top-p 0.95 --min-p 0.01 --top-k 40

That is giving me 31.5 t/s.

6 Reply

Educational-Agent-32 • 23h ago

May i ask what is MXFP4 is ?

1 Reply

UnknownLegacy • 15h ago

I am not 100% sure. But from my understanding it means:
Microscaling + FP4
MX is similar to when you see like Q4_K_XL. It's about as small as the Q models without much quality loss compared to a "Q" model of a similar size. It's also quite new and designed for hardware acceleration.
FP4 is 4-bit float, which is better quality than "Q" models, but generally larger in size and harder to run. However, "Blackwell" GPUs (the RTX5000 series) supports FP4 natively.

I was using the UD_Q4_K_XL model previously, but after reading that my GPU supports FP4 natively, I swapped. I just saw that OP and someone else in the thread was using "MXFP4" so I looked into it while trying to reproduce their ~23t/s.

2 Reply

Educational-Agent-32 • 14h ago

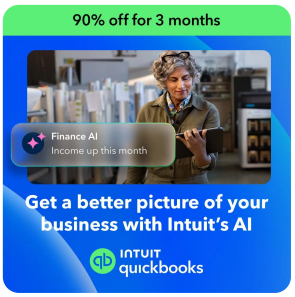
Wow thanks for these valuable information, i will try it on my 9070 XT while its supported

1 Reply

u/QuickbooksIntl • Promoted

A more powerful way to run your business

Sign Up quickbooks.intuit.com



bad_detectiv3 • 2d ago

bobaburger • 2d ago

it will. i'm getting pp 245 t/s tg 19 t/s on 5060 ti + 32gb ram

7 Reply

mrstoatey • 2d ago

What runtime and options are you using?

1 Reply

bobaburger • 2d ago

just default llama.cpp options

```
llama-server -m ./Qwen3-Coder-Next-MXFP4_MOE.gguf -c 64000 -fa 1 -np 1
```

3 Reply

bad_detectiv3 • 2d ago

Thanks I will try it over the weekend OP claim to be as good as Claude model for coding is hard to believe Last time I checked, so called Gemini flash was still a 200b model that Google provided instant response

1 Reply

bobaburger • 2d ago

not as good as claude, but if you are patient, you could get a decent results. i think this can be used as the last resort after you run out of quotas for other free usages.

1 Reply

iamapizza • 2d ago

> Qwen3-Coder-Next-MXFP4_MOE.gguf

Where did you download it from please?

1 Reply

bobaburger • 2d ago

it's here https://huggingface.co/unsloth/Qwen3-Coder-Next-GGUF/blob/main/Qwen3-Coder-Next-MXFP4_MOE.gguf

3 Reply

zerd • 1d ago

How much of a performance difference does MXFP4_MOE do vs UD-Q4_K_XL?

iamapizza • 2d ago

Running it in docker with CUDA

```
docker run --gpus all -p 8080:8080 -v /path/to/Models:/models ghcr.io/ggml-org
```

I'm getting about 23 t/s on 5080 TI + 32 GB RAM. Notice how I have much fewer arguments than OP.

4 Reply

social_tech_10 • 2d ago

Almost all those command line arguments are just the default values. Here it is with only the non-default options, and many of those options are also probably not needed as well:

- llama-server -m ../GGUF/qwen3-coder-next-mxfp4.gguf -t 12 -cmoe -c 131072 -b 512 --temp 1.0 --min-p 0.01 --host 0.0.0.0

By default

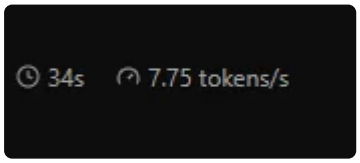
- -t (--threads) number of CPU threads to use during generation, default: -1 (automatic) - This should probably be set to automatic unless you specifically wan to use fewer CPU cores than you hav available
- -cmoe (--cpu-moe) keep all Mixture of Experts (MoE) weights in the CPU - Using the "--fit" command-line argument instead will automatically load as many many Experts into VRAM as will fit, and load the rest on the CPU.
- -c (--ctx-size) defaults to model training size, for that model, 256K - Leaving this as the default (with --fit) will give you the optimal context size for your system's RAM and VRAM
- -b (--batch-size) 2048
- --temp 0.80 - Increasing this setting to 1.0 increases "randomness" and "creativity", which might not be helpful for coding tasks.
- --min-p 0.05 (0.0 = disabled) - Solid research on this recommends settings between 0.05 and 0.1 Introducing Min-p Sampling: A Smarter Way to Sample from LLM, which makes me think this might be a misconfiguration based on bad advice, or perhaps a misplaced decimal point.

All things considered, the best command line for OP is probably just this:

- llama-server -m ../GGUF/qwen3-coder-next-mxfp4.gguf --fit --host 0.0.0.0

8 Reply

Juan_Valadez OP • 2d ago



Hahahashaha that was so funny sorry!

1 Reply

pmttyji • 2d ago
Top 1% Commenter

That's a good t/s for that config. What t/s are you getting for 256K context? It won't decrease t/s much.

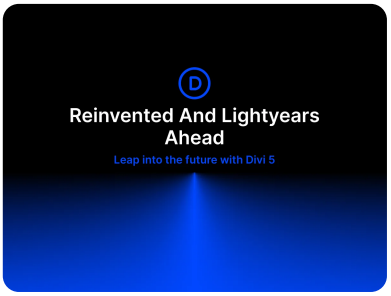
Also try -fit flags to see any good impact

2 Reply

u/DiviWordPress • Promoted

Divi 5 is lightyears ahead of Divi 4. There is simply no comparison. Experience the difference today!

Learn More elegantthemes.com



Reddit_User_Original • 2d ago

You have my exact system specs so i need to try this

2 Reply

Odd-Ordinary-5922 • 2d ago
Top 1% Commenter

I also have a 3060 12gb + 64gb ram. Try using --fit on its better than -cmoe

17 Reply

ABLPHA • 1d ago

Am I missing something? Every time I try --fit it stutters like crazy and eventually comes to a complete halt, and my DE barely stays alive. When I use --n-cpu-moe 47, it runs absolutely fine with long-context chats and the DE even has breathing room left. I'm running a larger quant but still, with a manual config I can actually squeeze out more out of my hardware than with --fit it feels like

2 Reply

DHasselhoff77 • 1d ago

Try --fit-target 512 or 1024 to leave some room for your desktop environment.

1 Reply

Odd-Ordinary-5922 • 1d ago
Top 1% Commenter

alenym • 2d ago

I'm really envy you 🤩

2 Reply

_bones__ • 1d ago

I'm getting about 13-16 tokens/s on a 3080 12GB. Not sure where the speed difference is from.

2 Reply

Hour-Hippo9552 • 2d ago

Sorry to ask d*b question I'm quite new to the scene. I just recently used local llm for personal hobby project and so far i'm liking it (with so many trial and errors finally found a good model for daily driver even for work). I'm interested to try Qwen 3 coder next but it says it is 80B and for q4_k_m it requires at least 40-50gb vram. HOw are you fitting it in 12gb? How's the performance? cpu/gpu temp? long session?

1 Reply

Odd-Ordinary-5922 • 2d ago
Top 1% Commenter

he said he has 64gb ram which lets him offload some layers to be computed to the cpu + ram, the performance will always be slower than a gpu but since Qwen3Coder only has 3B active parameters the speeds should still be decent.

2 Reply

Protopia • 2d ago

What is needed is an intelligent system that dynamically decides which layers or experts should be in GPU, and swaps them in and out from main memory cache as necessary to maximise performance.

2. If you had this, and the 3B active parameters were always running on the GPU, then the model should run entirely on (say) a 4GB consumer GPU.

3. Then you can try different quantizations to improve quality.

4. You can improve quality by optimising the context, and smaller context should also run faster. It's not just about the hardware, the model and the llamacp parameters.

1 Reply

Odd-Ordinary-5922 • 2d ago
Top 1% Commenter

if the active layers were swapped thousands of times in order to put the layers on the gpu then it would actually be slower as its too much compute

2 Reply

fixed, and CPU ram to vRAM transfer is reasonably fast, so losing the experts needed at the start of the call isn't going to be that much slower. But this is exactly how operating systems work - the more ram you have the less they swap things in and out from disk. The concept being better to run slowly than not run at all.

1 Reply

Odd-Ordinary-5922 • 1d ago
Top 1% Commenter

the experts change on a token to token basis so they arent fixed. Its only that 3b are active at all times

1 Reply

Protopia • 1d ago

Ah - since this is the case we can't swap them in and out. But I would imagine that there is some kind of optimisation that can be done to put the most likely and inference intensive ones in GPU, and the less likely and less intensive ones in normal memory with CPU inference.

1 Reply

More replies

Danmoreng • 2d ago

Windows or Linux? I get around 39 t/s with 5080 Mobile 16GB and 64GB RAM. 23 t/s seems a bit low, even if it’s just a 3060. Maybe I’m wrong though.

1 Reply

ArtfulGenie69 • 1d ago

When it's slower you can bet it's windows.

1 Reply

wisepal_app • 2d ago

thanks i will try this configuration. do you use it just in chat interface or with agentic coding tools like opencode etc?

1 Reply

guigouz • 2d ago

He said he's using Claude code

2 Reply

wisepal_app • 2d ago

Sorry, i missed that part.

2 Reply

Any estimate on t/s for 4090+128gigs ram?

1 Reply

timbo2m • 2d ago

Because it doesn't fit into vram there's a lot of back and forth over shuffling between ram/vram so it depends on other factors like cpu and bus.

For my 4090 in an i9 with 32GB RAM for the 4 bit quant my numbers are:

256k context = 24 tps

128k context = 26 tps

64k context = 27 tps

32k context = 28 tps

This was the exact settings (adjust context size to preference):

llama-server --host 0.0.0.0 --port 8080 -hf unsloth/Qwen3-Coder-Next-GGUF:MXFP4_MOE --ctx-size 32768 --temp 1.0 --top-p 0.95 --min-p 0.01 --top-k 40 --fit on

3 Reply

puru991 • 2d ago

Still decent. Thanks for sharing

1 Reply

BraceletGrolf • 2d ago

This is the type of content I love this sub for, thanks a lot

1 Reply

charmander_cha • 2d ago

Alguém saberia dizer qual melhor quantizacao para uma placa AMD?

1 Reply

TheCientista • 2d ago

Can I get similar performance from a 4070ti + 32GB DDR4?

1 Reply

jacktritus • 2d ago

I have a RTX 3060 12 GB but only 48 GB of RAM, should I try this?

How does Qwen3-coder-next compare to GLM-4.7-Flash-UD-Q4_K_XL.gguf

I've just setup OpenClaw in a docker container for isolation, using just webchat. GLM seems fine, but if Qwen3 is better, I'm all for it!

1 Reply

Amaria77 • 2d ago

Yeah. I have a 5070ti and a 4070 with 64gb of ddr4. I've been pretty impressed with qwen3-coder-next 80b q4km for basically everything I've thrown at it, even with half the model plus the kv cache (I also run ~128k) in my system memory. I mean, I'm not an expert by any means and am only giving it small chunks of work to do at a time, but it's been subjectively pretty capable. Though I'm going to have to give mxfp4 a shot looking at your results.

1 Reply

rm-rf-rm • 2d ago

Top 1% Commenter

Is it actually performing as well as Sonnet 4.6/Opus 4.6 to the point that you cancelled your subscription?

1 Reply

element-94 • 2d ago

There's no way its going to be a parity match. But for experienced engineers who can explain exactly what they want, I can see it working out.

6 Reply

Fresh_Finance9065 • 2d ago

See if --mlock, -kvu or --swa-full give you any performance boost

1 Reply

rorowhat • 2d ago

How big is this model?

1 Reply

mr_Owner • 2d ago

Try also cache ram 0 and k and v cache at q8.

1 Reply

sagiroth • 2d ago

Advice running this on 8gb vram and 32gb ram?

mircatmin • 1d ago

Excuse the ignorant question here. I'm struggling to get a feel for how quick 23 t/s is. Half as fast as sonnet 4.6? A tenth as fast?

Would a job on sonnet which takes 20 minutes take 24 hours at this speed?

1 Reply

nikolaiownz • 1d ago

Can you give me a quick guide to run this ? I only run lmstudio but I want to try this out

1 Reply

Help lead our community
Apply to be a moderator

Dismiss Apply

Reddit Rules Privacy Policy User Agreement Accessibility Reddit, Inc. © 2026. All rights reserved.