Speculative decoding tuning: Inkling 2 Mini on one RTX PRO 6000

26 November 2026 · 4 min read · blogs.iictp.org/speculative-decoding-inkling-2-mini-rtx-pro-6000-tuning

Short answer: speculative decoding raised single-user generation speed for the open Inkling 2 Mini model on one NVIDIA RTX PRO 6000 from about 157 to 190.6 tokens per second in a Hugging Face experiment published on 25 November 2026, after switching to a vLLM nightly build gave a first gain over the SGLang baseline. The author reports identical greedy outputs with and without speculation. The method of measurement is as useful as the result.

TL;DR The short version

- A Hugging Face experiment on 25 November 2026 raised Inkling 2 Mini's single-user speed on one RTX PRO 6000 from about 157 to 190.6 tokens per second with speculative decoding.
- Switching to a vLLM nightly build gave a first gain over the SGLang baseline, and greedy outputs were identical with and without speculation.
- The measurement method is as useful as the result: check output equality and measure on your own prompts.

Read online, listen to it, or change the text size: https://blogs.iictp.org/speculative-decoding-inkling-2-mini-rtx-pro-6000-tuning

What is speculative decoding?

A small drafting model proposes several next tokens and the main model verifies them together, keeping the ones it agrees with. When drafts are often accepted, fewer expensive steps are needed. Inkling 2 Mini is described as a looped mixture-of-experts model of about 26 billion parameters, 3 billion active per token, whose checkpoint bundles a drafter called DSpark.

How the experiment worked

- A coding agent followed an inference optimisation protocol on cloud GPU jobs.
- Measure hardware ceilings first.
- Change one variable at a time and run each candidate between two control runs.
- Keep a change only if it beats both controls without harming other metrics.
- Require long-context, concurrency and output-equivalence checks before choosing a winner.

What is new compared with copying a recipe

Published serving recipes for this model targeted other GPUs. The write-up shows how to tune an untested card without trusting single runs, and logs failures as well as wins. The final configuration uses seven draft tokens, probabilistic draft sampling, a 262,144 token maximum length and 0.85 GPU memory utilisation.

Results reported by the author

- 46 valid benchmark runs.
- Single-user decode: 136 to 190.6 tokens per second overall, up 40%.
- Eight-user aggregate: about 570 to 716 tokens per second, up 26%.
- Switching engines without speculation gave 157 tokens per second; speculation added about 21% on top.
- Greedy outputs were byte-identical on six prompts, and long-context retrieval passed three of three up to about 250,000 tokens.
- The estimated memory bandwidth ceiling with speculation was about 201 tokens per second, so the result was near 95% of it.

Limitations and open questions

- One model, one GPU, one prompt panel of six, at temperature 1.0.
- The build was a nightly image; behaviour can change, so the author pinned its digest.
- Byte-identical greedy output on six prompts does not prove equivalence in every case.
- The comparison between engines reflects their state on that date and card. Sources were read on 26 November 2026. Products, rules and standards change, so check the linked originals before you rely on any detail.

How to reuse the method

1. Write down the metrics: single-user speed, aggregate speed, prompt processing and time to first token.
2. Warm up across prompt lengths and concurrency before timing; the author saw compilation distort first measurements.
3. Bracket every change between controls and discard gains smaller than run-to-run noise.
4. Search the whole server log for memory errors, not only the last line.
5. Leave memory for the drafter: the new vLLM default let the cache claim the space it needed.
6. Verify outputs and long-context behaviour, then pin the image and model revision.

What it means for local and private LLM operators

On this setup, extra flag tuning had little headroom once bandwidth became the limit. Estimating that ceiling first tells you whether to keep tuning, change the drafter or change hardware. Acceptance rate depends on your prompts and temperature, so measure with your own traffic. Background: the Inkling 2 Mini architecture.

Frequently asked questions

- **Does speculative decoding change the answer?** Some methods preserve the output distribution; this run chose lossless settings and found identical greedy outputs on its checks.
- **How much faster was it?** About 21% on top of the same engine without speculation in this experiment.
- **Why did it fail at first?** The cache used the memory the drafter needed until limits were set.
- **Will I get the same numbers?** Not necessarily. Hardware, prompts, temperature and software version all matter.

Related reading

- Inkling 2 Mini explained
- vLLM 0.34 upgrade guide
- Hardware for local AI
- Local LLM Practitioner certification

Related credential: Local LLM Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. Niels Rogge, Hugging Face. Hill-climbing Inkling 2 Mini inference on one RTX PRO 6000. 25 November 2026. huggingface.co/blog/inkling-2-mini-inference-hillclimb

Published by IICTP, an independent technical credential body, on 26 November 2026. The current version of this post is at https://blogs.iictp.org/speculative-decoding-inkling-2-mini-rtx-pro-6000-tuning. More guides: blogs.iictp.org.