

I'm (mostly) picking models on speed now, not intelligence

August 2, 2026 · [Martin Alderson](#)

For the first time I can remember, I'm not choosing my daily driver models on raw intelligence. I'm choosing them on speed.

Have we reached an intelligence tipping point?

This is probably going to age like spoiled milk, but right now, models around the ~Opus 4.6 level seem to be 'smart enough' for most of my daily tasks - code, pulling together research, designing slide decks and doing analytical tasks against a plethora of databases.

While like most I was hyped to play around with Fable, ironically the US Gov shutdown gave everyone time to get used to Opus again. When Fable came back post-hype with additional guardrails, the first thing I noticed was just how *slow* it is^[1]. So slow, actually, that I switched back to Opus pretty quickly.

I've spent a *lot* of my career making software fast. It's remarkable how much better software feels to interact with when it's *fast*. In my experience (and many studies), you can take the most beautiful product, but if it's slow, you won't enjoy using it. Equally, you can take a very basic product that's super fast and it will feel brilliantly utilitarian.^[2]

It's clear to me that when only a few, big, slow models cleared the aforementioned (and hypothetical) intelligence bar, it wasn't worth the trade off really to use a slower model. Whatever speed you gain you quickly lose in having to redo it because it was broken.

Is 100tok/s the new 100ms?

I've [written before](#) about agents feeling like dialup, back when frontier models were crawling along at 30-60tok/s. That has changed faster than I expected.

The key fact I remember is that to humans, ~100ms feels 'instant', the gold standard. I reckon 100tok/s output on a model is about as fast as I can keep up with. After that, it comes in faster than I can (skim) read. This isn't an exact bar, because increasingly most of the model time is spent in reasoning, and not actually showing you output tokens. And also it massively varies on your output, prose gets output with far fewer tokens per character than code, so your mileage may (and will) vary.

But roughly, 100-200tok/s *to me* seems pretty damn fast. Below 50tok/s output feels increasingly slow. Ironically, going past 200tok/s seems almost unnerving - you can try a [model out here](#) at 10,000tok/s+ (!). I'm sure this feeling will edge upwards as we get used to it and push our agents to do more complicated work.

Given the plethora of new models that I think are ~clearing the aforementioned bar - such as GLM5.2 and DeepSeek V4 Flash GA - that are open weights *and* small(er), we now have a wide range of models and speed. If you look at the speed rankings of various providers for GLM5.2 on [OpenRouter](#) you can see the enormous range of serving speed - from less than 30tok/s at the bottom to 129tok/s at the top.

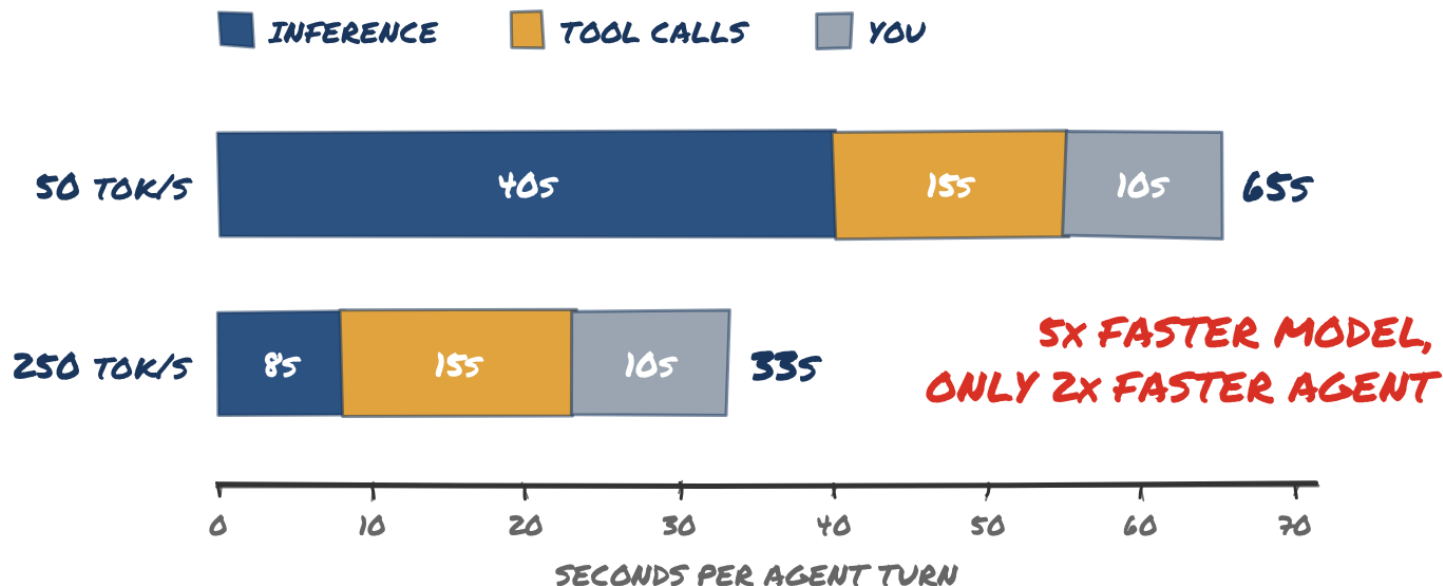
This is another huge plus to the open weights ecosystem. While there are great benefits in cost that are obvious, the fact that providers are also incentivised to compete on speed like this is really interesting.^[3]

But there are limitations

If you're familiar with [Pareto's Principle](#) and [Amdahl's Law](#) you'll know what's coming up.

Assuming "good enough" models continue to get faster and faster, increasingly the speed benefit is lost to tool calls, and us humans overseeing them.

Take an agent using a model processing at 50tok/s. Most of the time is spent waiting for inference to come back. Now run the same turn at 250tok/s and you'll see that increasingly you are bottlenecked on tool calls on your "local" machine and your decision making.



Rough numbers, but the shape holds. The 5x speedup on the model only buys you a 2x speedup on the turn, because the other 25 seconds didn't move.

And *even worse*, making your local machine faster on these tool calls is sort of stalling out, because hardware costs have gone parabolic *because* of AI. Yet again another weird derivative effect of the AI market.

So I suspect (for now at least) there is a limit to how much demand there will be for speed, past a certain point. No doubt there'll be some examples where huge amounts of reasoning are useful (like mathematics research), and speeding that up is helpful. But I'd expect many agents to start getting bottlenecked on your local/internal hardware, database calls and other bits of latency.

The price war is coming

Interestingly OpenAI reduced the cost of their Luna variant by 80% just before the DeepSeek V4 Flash GA release, making it remarkably affordable for a frontier model. While I haven't had as much luck with getting great output out of it vs GLM5.2, I think it points towards an absolute bloodbath of pricing at this end of the market.

You can see this happening on OpenRouter with GLM5.2 - endless discounts being offered to try and attract customers in. We're already down to \$0.42/\$1.32/MTok on GLM5.2 - 5% of the price of Opus.

Standard ▾

Latency / throughput

P50 ▾

Filter quantization ▾

Provider ↕		↕ Input /M	↕ Output /M	↕ Cache Read /M	↕ Latency	↕ Throughput	↕ Uptime
 StreamLake	 70% off	\$1.40 \$0.42	\$4.40 \$1.32	\$0.26 \$0.078	1.97s	29 tps	 99.59%
 NovitaAI	 49% off	\$1.40 \$0.7168	\$4.40 \$2.253	\$0.26 \$0.1331	1.67s	36 tps	 99.80%
 Decart	 40% off	\$1.20 \$0.72	\$3.00 \$1.80	\$0.20 \$0.12	1.38s	67 tps	 99.86%
 DeepInfra		\$0.75	\$2.40	\$0.14	0.66s	109 tps	 99.13%
 Baidu Qianfan	 46% off	\$1.40 \$0.756	\$4.40 \$2.376	\$0.26 \$0.1404	1.38s	45 tps	 98.61%
 CoreWeave		\$0.76	\$2.42	\$0.14	0.60s	115 tps	 99.36%
 AkashML	 45% off	\$1.40 \$0.77	\$4.40 \$2.42	\$0.26 \$0.143	1.74s	49 tps	 88.02%
 GMICloud	 34% off	\$1.40 \$0.924	\$4.40 \$2.904	\$0.26 \$0.1716	1.49s	40 tps	 99.05%
 Inception		\$0.94	\$2.90	\$0.17	0.54s	29 tps	 98.72%

While the very cheapest is slow, for not much more you can get 109tok/s from DeepInfra.

As the next generation set of GPUs start being deployed over the next few months - Nvidia's Vera Rubin series and AMD's MI400s, amongst others - the new HBM4 memory in those chips will deliver a 2x+ speedup on output tokens from memory bandwidth alone, plus more on top from additional compute and interlink.

In 2027 it's very possible we'll have very good quality models, at reasonable prices running at 500tok/s+. Staring at "still thinking on xhigh effort" for most of your day may finally become a thing of the past.

What will be interesting to watch for - and I'm not sure where to bet - is if the vast 2-3T+ param models actually do perform *dramatically* better for everyday tasks. On one hand it feels like we've hit a sweet spot right now, on another having an order of magnitude more intelligence in the model may make that sweet spot look very, very primitive.

1. Second of course is the endless guardrails firing, which tend to happen at the worst possible time - just when I'm getting deep into a difficult task and I feel I could do with the extra "firepower" that Fable offers, but that's a story for another day. 

2. A classic example is something like Craigslist or Hacker News. While they look dated, they are so damn responsive you don't notice. Equally, your "standard" SPA app serving 30MB of React to render a homepage feels like treacle and a chore to use most of the time, despite what was surely an enormous spend on design and product. [!\[\]\(83eb2aa26b610eb6a9dca7cf4702d681_img.jpg\)](#)
3. I'm aware that both OpenAI and Anthropic have offered fast variants of models for a long time, but the API pricing is eye watering. Unless you are tokenmaxxing your benchmarks with a blank cheque, I haven't come across anyone that uses them for day to day operation. Having great models that are super fast at a reasonable price is a very recent addition to the market. [!\[\]\(94dfacbf937cdd7da4837a6fcd8fc785_img.jpg\)](#)

RELATED

[The Coming AI Compute Crunch](#)

[Are we in a GPT-4-style leap that evals can't see?](#)

[Why Claude's new 1M context length is a big deal](#)

[← Back to Home](#)