CATALOGUE LATENCY · NOVEMBER 2026

Model latency, measured.

A 1,828-cycle synthetic probe of 39 catalogue models over six days, measured on Plugsky production infrastructure.

16–22 NOV 2026 · UTC

1,828	71,292	39	10
PROBE CYCLES	MODEL CHECKS	CATALOGUE MODELS	PROBE STREAMS

METHOD IN BRIEF

Every five minutes, a production cron sends one short chat probe to each distinct upstream. The result is recorded against every Plugsky alias that shares that upstream. Latency is successful-probe time; failures and rate limits are reported separately.

INSIDE 02 Method · 03 Latency by model · 04 Probe streams · 05 Plan economics · 06 Limits & sources

Method and data provenance

This is an operational latency measurement, not a laboratory benchmark. It replays the same synthetic probe Plugsky runs in production to detect broken upstreams, across the seven retained health-log segments.

THE PROBE

- Request. A single non-streaming chat completion with the user message “Reply OK.” and max_tokens = 128. Reasoning is disabled where the upstream supports it.
- Cadence. A cron job every five minutes; 1,828 logged cycles in this window.
- Scope. 39 catalogue aliases. One probe per distinct upstream per cycle; aliases sharing an upstream are served from a per-cycle cache and record the same result.
- Timing. Wall-clock time from request start to a valid response, including any key rotation or retry inside the probe’s 45-second budget. A successful probe can include a retry.
- Transport. Server-side HTTPS from Plugsky production to the upstream inference API. It does not include a customer’s network path or Plugsky’s public edge.

ONE PROBE, MANY ALIASES

> The 39 aliases produce only 10 distinct status-and-latency signatures across all 1,828 cycles — consistent with one probe per upstream and alias sharing. We call these probe streams. In a typical cycle the 39 log lines contain six distinct latency values (range 5–7). Treat the stream as the independent unit, not the alias; definitions are on page 4.

STATUS VOCABULARY

Status	What it means in the log	Checks	Share
ok	Valid response	69,059	96.87%
rate_limited	HTTP 429/503 after retries; transient	1,799	2.52%
slow	First consecutive genuine failure (soft-fail)	326	0.46%
broken	Second or later consecutive genuine failure	108	0.15%
Total	All model checks in the window	71,292	100.00%

“slow” is a reliability state, not a latency threshold. The printed summary folds **rate_limited** into its “slow” count; this report keeps the per-check statuses. A **slow** or **broken** line is a failed probe, not a measured slow success.

PERCENTILES AND WINDOW

Percentiles. p50, p95 and p99 are nearest-rank percentiles over successful (**ok**) probes only. **ok_pct** is over all checks. Failed and rate-limited timings are excluded from latency; they remain visible in **ok_pct** and in the CSV notes.

Window. The logs carry no per-line timestamps. The window is an outer bound taken from file mtimes: the oldest parsed segment begins after **health.log.7.gz** was last written at 2026-11-16 00:05 UTC, and **health.log** was last written at 2026-11-22 08:40 UTC. 1,828 cycles at a nominal five-minute cadence fit inside that 6 d 8 h 35 m span.

HOW THE LOG READS

```
  plugsky-micro              ok          374ms
  plugsky-embed              ok           64ms

Summary: 39 checked, 39 ok, 0 slow, 0 broken
```

Blocks are split on the **Summary:** line. Each block contains 39 model lines; a line carries model, status and latency. ANSI colour codes are stripped before parsing. 1,828 blocks were parsed from seven files.

Latency by catalogue model

All 39 entries, ranked by median successful-probe latency. Every entry has 1,828 checks across the same window. Stream codes are defined on the next page; aliases in one stream share a single upstream probe and therefore return identical values.

#	Catalogue model	Stream	OK %	p50	p95	p99	Mean
1	plugsky-embed	S1	100.00	77	112	181	79.1
2	plugsky-embed-multilingual	S1	100.00	77	112	181	79.1
3	plugsky-embed-nim	S1	100.00	77	112	181	79.1
4	plugsky-translate	S2	100.00	164	426	882	205.1
5	plugsky-parse	S3	96.72	439	1,927	9,782	848.4
6	plugsky-gemma3-nano-2b	S4	99.34	444	1,968	9,528	852.4
7	plugsky-gemma3-nano-4b	S4	99.34	444	1,968	9,528	852.4
8	plugsky-gpt-oss-20b	S4	99.34	444	1,968	9,528	852.4
9	plugsky-lightning	S4	99.34	444	1,968	9,528	852.4
10	plugsky-lite	S4	99.34	444	1,968	9,528	852.4
11	plugsky-micro	S4	99.34	444	1,968	9,528	852.4
12	plugsky-mistral-small	S4	99.34	444	1,968	9,528	852.4
13	plugsky-nano	S4	99.34	444	1,968	9,528	852.4
14	plugsky-nano-omni	S5	99.34	444	1,968	9,528	852.5
15	plugsky-phi	S4	99.34	444	1,968	9,528	852.4
16	plugsky-plus	S5	99.34	444	1,968	9,528	852.5
17	plugsky-qwen-next	S4	99.34	444	1,968	9,528	852.4
18	plugsky-tiny	S4	99.34	444	1,968	9,528	852.4
19	plugsky-gemma-4	S6	98.63	536	1,074	4,766	707.1
20	plugsky-qwen-vl	S6	98.63	536	1,074	4,766	707.1
21	plugsky-vision-fast	S6	98.63	536	1,074	4,766	707.1
22	plugsky-coder-fast	S7	94.47	586	2,350	5,987	931.1
23	plugsky-deepseek-flash	S7	94.47	586	2,350	5,987	931.1
24	plugsky-gpt-oss	S7	94.47	586	2,350	5,987	931.1
25	plugsky-coder	S8	94.42	588	2,389	6,393	958.7
26	plugsky-coder-ultra	S8	94.42	588	2,389	6,393	958.7
27	plugsky-deepseek-pro	S8	94.42	588	2,389	6,393	958.7
28	plugsky-frontier	S8	94.42	588	2,389	6,393	958.7
29	plugsky-gpu-27b	S8	94.42	588	2,389	6,393	958.7
30	plugsky-kimi	S8	94.42	588	2,389	6,393	958.7
31	plugsky-llama4	S8	94.42	588	2,389	6,393	958.7
32	plugsky-longctx	S8	94.42	588	2,389	6,393	958.7
33	plugsky-max	S8	94.42	588	2,389	6,393	958.7
34	plugsky-pro	S8	94.42	588	2,389	6,393	958.7
35	plugsky-reasoning	S8	94.42	588	2,389	6,393	958.7
36	plugsky-thinker	S8	94.42	588	2,389	6,393	958.7
37	plugsky-ultra	S8	94.42	588	2,389	6,393	958.7
38	plugsky-minimax	S9	93.82	599	3,146	9,822	1,108.7
39	plugsky-mistral-medium	S10	89.06	3,842	28,366	39,871	7,996.8

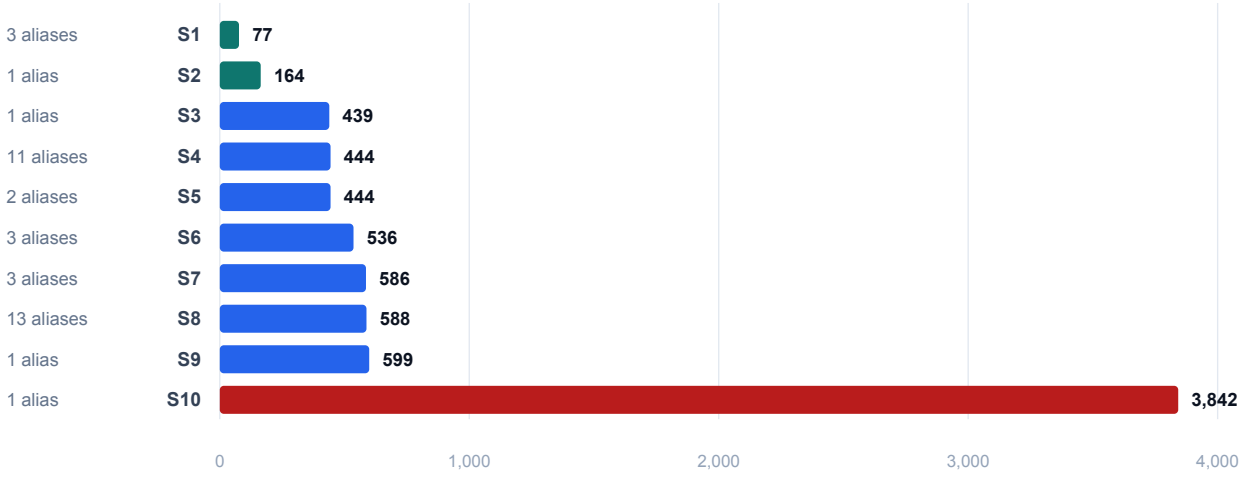
Milliseconds unless noted. p50/p95/p99: nearest-rank over successful probes. Mean: arithmetic mean over successful probes. OK %: successful probes divided by all 1,828 checks. Rows with identical values are aliases of one upstream probe, not independent measurements.

Ten distinct probe streams

The catalogue is wider than the measurement: 39 aliases produce 10 distinct probe streams, identified by identical status-and-latency signatures across all 1,828 cycles. Aliases in a stream are not independent observations — one upstream call feeds every alias in that stream.

Stream	Aliases	Representative catalogue entries	p50	p95	p99	OK %
S1	3	embed, embed-nim, embed-multilingual	77	112	181	100.00
S2	1	translate	164	426	882	100.00
S3	1	parse	439	1,927	9,782	96.72
S4	11	micro, lite, tiny, phi, qwen-next +6	444	1,968	9,528	99.34
S5	2	nano-omni, plus	444	1,968	9,528	99.34
S6	3	gemma-4, qwen-vl, vision-fast	536	1,074	4,766	98.63
S7	3	coder-fast, deepseek-flash, gpt-oss	586	2,350	5,987	94.47
S8	13	coder, coder-ultra, max, pro, ultra +8	588	2,389	6,393	94.42
S9	1	minimax	599	3,146	9,822	93.82
S10	1	mistral-medium	3,842	28,366	39,871	89.06

p50 latency by probe stream (ms, successful probes)



- Speed floor. Embeddings are the fastest stream at 77 ms p50 (112 ms p95); translate is 164 ms p50. Both were 100% ok across all 1,828 checks.
- Chat cluster. Most chat streams sit between 444 and 599 ms p50. The exception is mistral-medium at 3,842 ms p50 and 28,366 ms p95.
- Rate limiting is concentrated. S8's 92 rate-limited cycles appear against 13 aliases (1,196 of the 1,799 rate_limited log entries), and S7 adds 273. 112 of 1,828 cycles had 10 or more non-ok aliases; the worst had 34 of 39 aliases non-ok.
- Reliability is uneven. S1 and S2 were 100% ok. mistral-medium was 89.06% ok with 91 slow, 19 broken and 90 rate_limited checks.

Plan economics

Published prices and fair-use limits for the paid tiers, captured from the live pricing page. These are plan ceilings, not measured throughput.

Plan	Monthly	Annual	Requests/day	Tokens/day	Concurrent	RPM	Max input/output
Hobby	$8	$76.80	1,000	2M	4	120	64K / 8K
Starter	$20	$192	2,000	5M	6	240	128K / 16K
Builder	$60	$576	Unlimited — no token cap	Unlimited — no token cap	10	600	256K / 32K
Scale	$120	$1,152	Unlimited — no cap	Unlimited — no cap	16	1,200	512K / 32K

Source: **plugsky.com/pricing**, retrieved 2026-11-22 08:41 UTC (Last-Modified 2026-11-22 08:40:12 UTC). Prices are per workspace list prices; annual billing is 20% off. First-month launch promotions are excluded. Daily limits reset at 00:00 UTC.

READING THESE NUMBERS TOGETHER

- The latency table is single-request. It does not exercise the concurrency column, queueing, or saturation.
- The probe rate is low. One request per upstream every five minutes is far below every paid plan’s requests-per-minute ceiling.
- Daily caps differ by tier. Hobby and Starter publish daily request and token caps; Builder and Scale are listed as unlimited with no token cap.
- Catalogue access. The pricing page says Free uses the Plugsky AI engine, while paid plans use the full model catalogue measured here.

APPLYING THE MEASUREMENTS

- Embeddings and translation are the speed floor. At 77 ms and 164 ms p50 they are the fastest entries in the catalogue; daily caps and concurrency, not model speed, are likely to bind first.
- General chat clusters around half a second. Most chat streams are 444–599 ms p50. Hobby’s 4 concurrent requests and 1,000 requests/day are the constraints to check for multi-user use.
- Long context is not covered by these timings. Builder and Scale are the only tiers above 128K input; the probe did not test long context, so these figures do not predict it.

If the workload is…	Published limits to check	Start with
Solo prototype, low volume	1,000 requests and 2M tokens per day	Hobby
Small team, moderate volume	2,000 requests and 5M tokens per day	Starter
Production app, more concurrency	10 concurrent, 600 RPM	Builder
Largest context, highest concurrency	512K input, 16 concurrent, 1,200 RPM	Scale

Positioning aid only, not a benchmark result. Choose on caps, context and concurrency; the latency medians here describe one short request at a time.

> Do not convert these medians into a throughput estimate. A plan’s concurrency limit, queueing, prompt length and output length all change the result. None of those were tested in this benchmark.

Limits, caveats and sources

WHAT THIS REPORT DOES NOT MEASURE

- No competitor latency, price or reliability. No competitor system was contacted or measured. Every number here is Plugsky catalogue data.
- No throughput under load. The probe is serial and single-request; it does not test concurrency, queueing, or saturation.
- No long-context timing. Every probe is a short prompt with a minimal answer; context length is not varied.
- No quality measurement. A valid response is not judged for correctness, reasoning, tool use, or safety.
- No streaming or time-to-first-token. The probe is non-streaming and records total response time.
- No customer-side latency. The probe runs server-side and excludes your network path and Plugsky edge overhead.
- No regional, time-of-day, or cold-start breakdown. The log has no timestamps, so cycles cannot be attributed to hours.
- No SLA or uptime guarantee. One probe every five minutes is a monitoring sample, not a census.

DATA-QUALITY NOTES

- 39 aliases, 10 probe streams. The 71,292 checks are not 71,292 independent observations.
- “slow” and “broken” are consecutive-failure states, not latency buckets. The printed summary also folds rate_limited into “slow”.
- The window is an outer bound derived from file mtimes. The logs have no per-line timestamps, and two cron firings were skipped while a prior run held the lock.
- The retained files cover about six days, not the full month of November. Older rotated segments exist on the server but were outside this report’s scope.
- Watchdog upstream overrides were recorded for mistral-medium and minimax during the window; their series may mix more than one upstream.
- Percentiles exclude failed and rate-limited timings. A fast 429 does not enter the latency distribution.

DATED SOURCES

Source	Location	Date / detail
Health logs	backend/storage/logs/health.log + .1–.6	1,828 cycles; last writes 2026-11-17 00:15 to 2026-11-22 08:40 UTC
Window boundary	backend/storage/logs/health.log.7.gz	mtime 2026-11-16 00:05:11 UTC; used as boundary only, not parsed
Pricing	https://plugsky.com/pricing	retrieved 2026-11-22 08:41 UTC; Last-Modified 2026-11-22 08:40:12 UTC
Probe method	backend/bin/health-check.php; ModelHealthProbe.php	source mtimes 2026-11-11 12:43 and 2026-10-19 11:19 UTC

File mtimes (UTC): health.log 22 Nov 08:40:06; .1 22 Nov 00:15:09; .2 21 Nov 00:30:10; .3 20 Nov 00:10:08; .4 19 Nov 00:00:24; .5 18 Nov 00:40:18; .6 17 Nov 00:15:07.

REPRODUCTION

benchmark.csv contains one row per catalogue alias, with the stream code and per-status counts in **notes**. README.md gives the exact parsing and percentile recipe. The PDF was built with reportlab 3.6.13. No figure in this report is modelled, estimated, or sourced from a third party.

> Report version 1.0 · 22 November 2026. Scope: Plugsky catalogue only; no competitor systems were measured. Window: 16–22 November 2026 UTC. Every figure can be recomputed from **benchmark.csv** and the retained production log set.

© 2026 Plugsky. Operational measurement, provided as-is. Prices and limits may change; check plugsky.com/pricing before purchase.