

Local AI: your team, your capacity

Four Qwen/Swift models · two R9700 GPUs · no provider token quota

174.7 tokens/s per response · 855.5 aggregate tokens/s with eight requests · ~128K input tokens. Practical capacity for assistants, long documents and parallel agents. Organise the server around your workload, without buying API credits or waiting for a provider quota to reset.

855.5

aggregate tokens/s · 8 requests submitted

174.7

tokens/s · one response (C1)

0 USD

local API charges

Four models, eight requests submitted together: wall throughput (tokens/s)



Median [min-max] of two repetitions; circle R1, square R2. Actual 128K input · low · output ≤ 4096 · window 262144.

A | Qwen3.8-27B / Paition | 2 R9700, 2 x TP1 | W3A4 / W3Rot | DFlash7 | KV KV4 | build 245c71d5

B | Swift1.5 Qwen3.8-27B / Paition | 2 R9700, TP2 | Quark RTN-MXFP4 | MTP3 (BF16 head) | KV FP8 | build 245c71d5

C | Qwen3.8 Flash Next original / Radiance native | 2 R9700, TP2 | IQ4R experts + INT8 trunk | MTP3 | KV FP8 | build ef8cecc8

D | Swift1.5 Flash Next / Radiance native | 2 R9700, TP2 | IQ4R experts + INT8 trunk | MTP3 | KV FP8 | build ef8cecc8

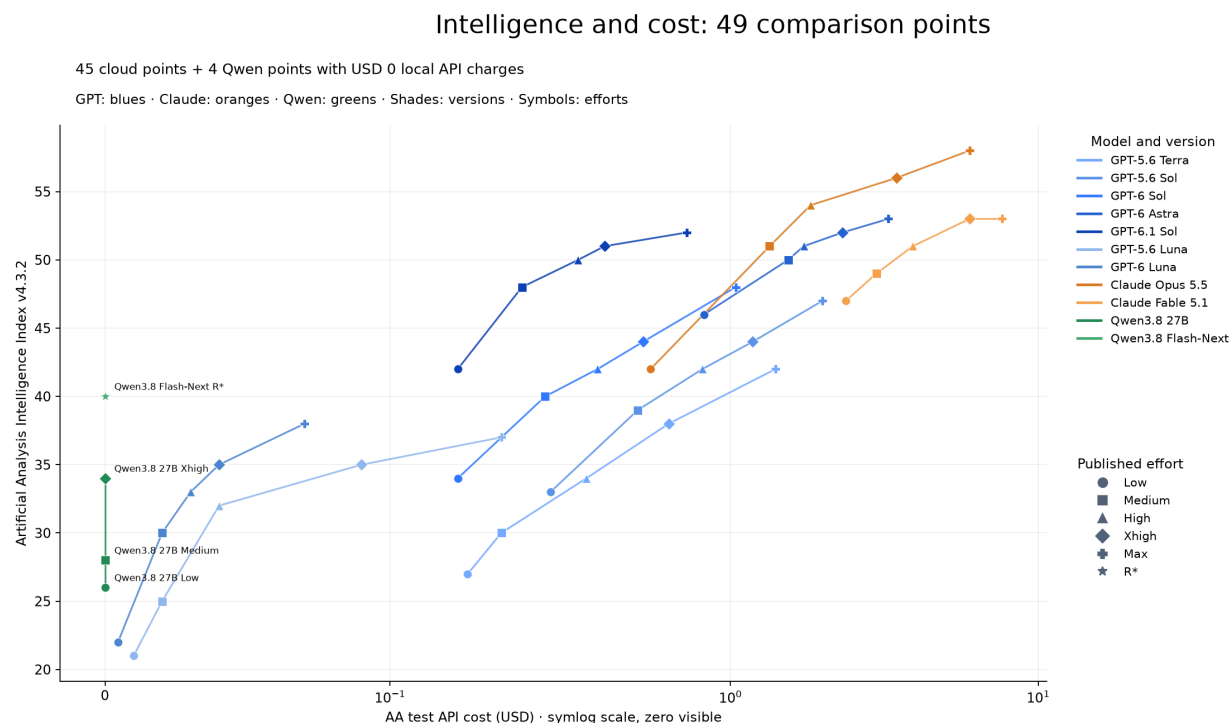
Two R9700 GPUs · input 127632-127674 tokens · natural cache. USD 0 = API tokens only; hardware/electricity excluded.

How many people? With Swift Flash Next at ~128K / low: one task, **174.7 tokens/s per response**; two tasks, **269.9 aggregate tokens/s**; four, **463.7**; eight, **855.5**, with **110.4 tokens/s per response**. Eight people each submitting one request correspond to the measured C8 submission point; a larger team can share the server with intermittent usage. Queueing and response times depend on load.

384 successful groups, 1,440 requests, two repetitions: Qwen3.8-27B, Swift1.5 Qwen3.8-27B, Qwen3.8 Flash Next and Swift1.5 Flash Next. Configured window: 262,144; actual measured input: up to ~128K; output $\leq 4,096$. Aggregate throughput includes prefill and queueing; generation speed per response excludes them.

Intelligence: position local models against the cloud

Local deployment can combine the quality of open models with zero API token charges. Compare reasoning levels and cost: all 49 points in the original chart are retained, including 45 cloud references and four Qwen points.



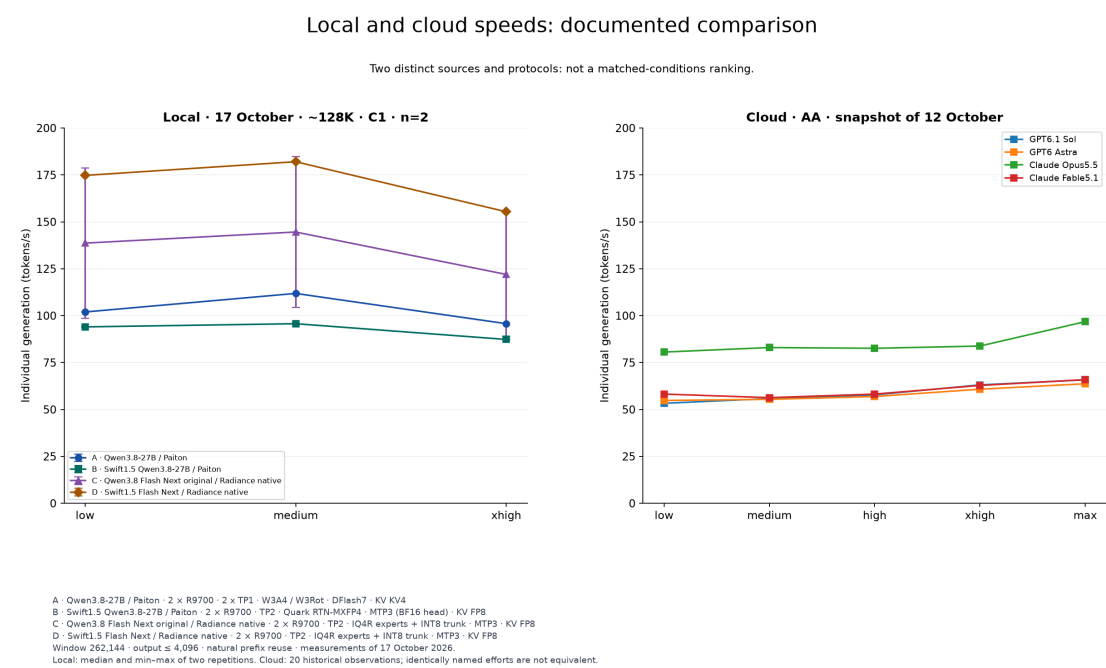
Source: Artificial Analysis, snapshot of 12 October 2026. Reference model scores; quantized recipe quality not remeasured.
AA test cost is not a per-million-token price. Electricity and hardware excluded from local API charges.

Numerical reference points: in the 12 October Artificial Analysis archive, Qwen27 scores **26 / 28 / 34** at low / medium / xhigh; Qwen Flash Next reaches **40** in the published setting. In the LiveBench selection, overall scores are **76.19 for Qwen Flash Next**, **75.27 for Qwen27** and **75.67 for Claude Sonnet 5.5 max**. The highest-ranked cloud models remain above them: Opus 5.5 max **83.22**, GPT-6 Astra max **82.16**. These distinct indices cannot be added, and rankings depend on the task.

Reading the chart: higher = better index; further right = higher API cost per Artificial Analysis task. Local API cost is **USD 0**. Scores describe published reference models; the quality of the local quantized and Swift variants was not measured here. Sources: [Artificial Analysis](#) and [LiveBench](#). Details, coding tests, GPQA and complete tables remain in the [base document](#).

Speed and cost: a practical local alternative

174.7 tokens/s per response at ~128K / low / C1, with up to eight requests submitted together: size the server around your workload. The chart retains all 20 cloud references and local speeds, with their distinct protocols.



Example: 128,000 input tokens + 4,096 output tokens	API charges, USD
Four local configurations	0
GPT-6.1 Sol	0.29696
Claude Opus 5.5	0.59392
GPT-6 Astra	1.4848

USD 0 means zero purchases of local API tokens, excluding hardware and electricity. Cloud examples use public prices archived on 18 October 2026, with no cache, tools, taxes or discounts. External cloud measurements and the local campaign use different tasks and contexts: the chart positions their orders of magnitude. Pricing sources: [OpenAI](#) and [Anthropic](#). All conditions, repetitions, cost comparisons and sources remain in the [complete document](#).