

Opus 5.5 Clears a Stalled Port; the Assembly Speedup Has a Narrow Benchmark

Coding Agents Alpha Tracker

2026-09-26

Opus 5.5 Clears a Stalled Port; the Assembly Speedup Has a Narrow Benchmark

By Coding Agents Alpha Tracker • September 26, 2026

Two practitioners report Opus 5.5 completing major language translations, while the linked `ttfx` PR gives a testable, decrypt-specific speed result rather than a general coding benchmark. The brief pairs that signal with concrete practices for scoping long runs, verifying agents, and evaluating new developer tools.

TOP SIGNAL

Long-horizon code translation is the strongest signal. Theo says Opus 5.5 got his four-month, on-and-off TypeScript-compiler-to-Rust port working in 10 hours, after GPT-5.6 Sol and GPT-6 Astra plateaued at roughly 35% and 85% of tests passing. [1, 2] DHH separately says Opus 5.5 translated Omarchy's `ttfx` from Rust to x86-64 assembly and claims up to $17\times$ faster. The linked PR reports $14.5\times$ on one `decrypt` test ($371\rightarrow 25.5$ ms) and separately lists 32.5 ms versus Rust's 371 ms, with 206/206 oracle cases matching—strong project-specific evidence, not a general coding benchmark. [3, 4]

TRY THIS

- **Give long runs a finish line and a quality target.** Theo's exact request: `/goal finish the port and make it faster`. Peter Steinberger used `/goal` with Astra to move OpenClaw off synchronous SQLite after it became limiting with one agent running 50 sessions in parallel; he reports 575 PRs landed so far, with improvements shipped along the way. [2, 5]
- **Make review adversarial and isolated.** Steinberger's auto-review skill starts a fresh-context reviewer, feeds its findings back to the writer, and repeats; then the agent gets a separate test box to install the app, inspect

the UI, and verify it end to end. He says this can turn a 10-minute PR into two hours, but lets him parallelize about 20 tasks—reserve the extra validation for changes where it pays back. [6]

- **Turn Slack bug reports into repro-to-PR jobs.** Boris Cherny says Claude Tag writes more than 50% of his PRs each day; his prompt is: “@Claude from now on, attempt to repro every bug in this channel end to end, running the full app. Once you reproduce the bug, put up a PR to fix it and tag the right team for code review.” [7] Claude Tag’s personal connectors are rolling out on Team plans, with Enterprise to follow; those personal connectors are for user-requested work, not scheduled or agent-started runs, which require connectors attached to the channel. [8]
- **Write the HLDD after the implementation works.** Geoffrey Huntley’s pattern: use pre-commit hooks and agent skills to keep the high-level design doc updated, then self-host it in the product. Put the human-authored rationale and success measures first, implementation details below, and editor remarks where a subsection needs its intent preserved. [9]

WHAT SHIPPED

- **Antigravity 2.0 adds /plan in the app and CLI.** The workflow explores files and dependencies without writing code, asks clarifying questions, then produces a reviewable plan with verification checkpoints. Review or adjust it and click **Proceed**; the docs say whether the agent pauses for approval is configurable through the artifact review policy, so set that explicitly if you need a hard gate. [10, 11]
- **Microsoft Copilot Code is an app builder, not the new default coding IDE.** It turns natural-language requests into dashboards, automations, and internal apps in a sandbox that can be hosted in a tenant. Frontier rollout is slated for month-end, Managed Runtime is already in preview, and Microsoft says developers will keep using GitHub Copilot for day-to-day work. [12]
- **Autopilot is a separate persistent agent.** Microsoft describes a cloud-hosted agent that watches channels, follows up, handles recurring work, and can resume projects days later; its blog says private-preview expansion is at month-end. Omar Shahine says it is built on OpenClaw, whose team worked with Microsoft on deployment readiness; local inference, file transfer, and code mode on machines connected to the OpenClaw gateway are still described as upcoming. [12, 13, 14, 15]
- **Cua Driver is now stable and open source for Omarchy.** TryCua says its native synthetic cursor for Hyprland enables OS-level multi-cursor control. The repo shows two sessions operating Calc and Inkscape on Omarchy and supports agent connections through CLI, MCP, or typed SDKs. [16, 17]

- **OpenRouter introduced typesafe/jev-router, a cache-aware model-and-reasoning-effort selector.** Theo’s objection is coding-specific: Jev categorizes but cannot see the codebase or the agent’s tools, so prompt-only complexity estimates may miss the work involved. He also doubts mid-task downgrades save much when cache writes are a large cost, and says he is still gathering numbers; his early comparison to a “DeepSeek 4.1 Flash router” is only an impression. [18, 19, 20, 21]

GO DEEPER

- **Video — Peter Steinberger on coding-agent review loops:** The useful segment is his fresh-context reviewer and isolated test-box workflow,



not a model demo. [6]

Peter Steinberger: Stop Sending Me Code. Send Me Your Prompt (34:00)

- **PR — Omarchy ttfx Rust-to-assembly port:** Inspect how Rust remains the front end and fallback for unported effects, while the assembly engine is checked against byte-for-byte oracle cases. [4]
- **Repo — Cua Driver:** The quickstart connects an agent to Calculator, asks it to compute 6×7 , and checks that the app displays 42; the repo also documents CLI, MCP, and SDK integrations. [17]

Editorial take: The edge is not simply “let it run”: specify the finish line, then give an independent reviewer and a real execution environment a chance to falsify the result. [2, 6]

Sources

1. X post by @theo
2. X post by @theo
3. X post by @dhh
4. Add an x86-64 assembly engine for all 37 effects by dhh · Pull Request #35
5. X post by @steipete
6. Peter Steinberger: Stop Sending Me Code. Send Me Your Prompt
7. X post by @bcherny
8. Claude Tag now supports personal connectors in channels
9. X post by @GeoffreyHuntley
10. X post by @antigravity
11. Plan (/plan)
12. Introducing the new Copilot with Home, Code and Autopilot - The Official Microsoft Blog
13. X post by @OmarShahine
14. X post by @steipete
15. X post by @steipete
16. X post by @trycua
17. trycua/cua: Scale computer-use 2.0 with open-source drivers, cross-OS fleets, and benchmarks for training, evaluation, and data generation.
18. X post by @OpenRouter
19. X post by @theo
20. X post by @theo
21. X post by @theo