Parallel Evaluation in n0is: Throughput, Latency, and Replication

Tyler Garrett / 2026-11-08 / Working paper 0.2

Abstract

Parallel workers can finish a test batch sooner even when individual tests take longer. We measured this tradeoff while evaluating small language models that drive interactive ETL workflows in n0is. Across 12 randomized blocks and 384 completed chains, four workers delivered 45.75 chains per minute against 29.16 for one worker, an estimated gain of 56.87%. Three and four workers had similar throughput within the uncertainty of the study. An independent experiment also found little evidence of a practical gain from a fifth worker. The measurements use local processes sharing one Mac GPU. The project also used four isolated Linux virtual machines for parallel test capacity.

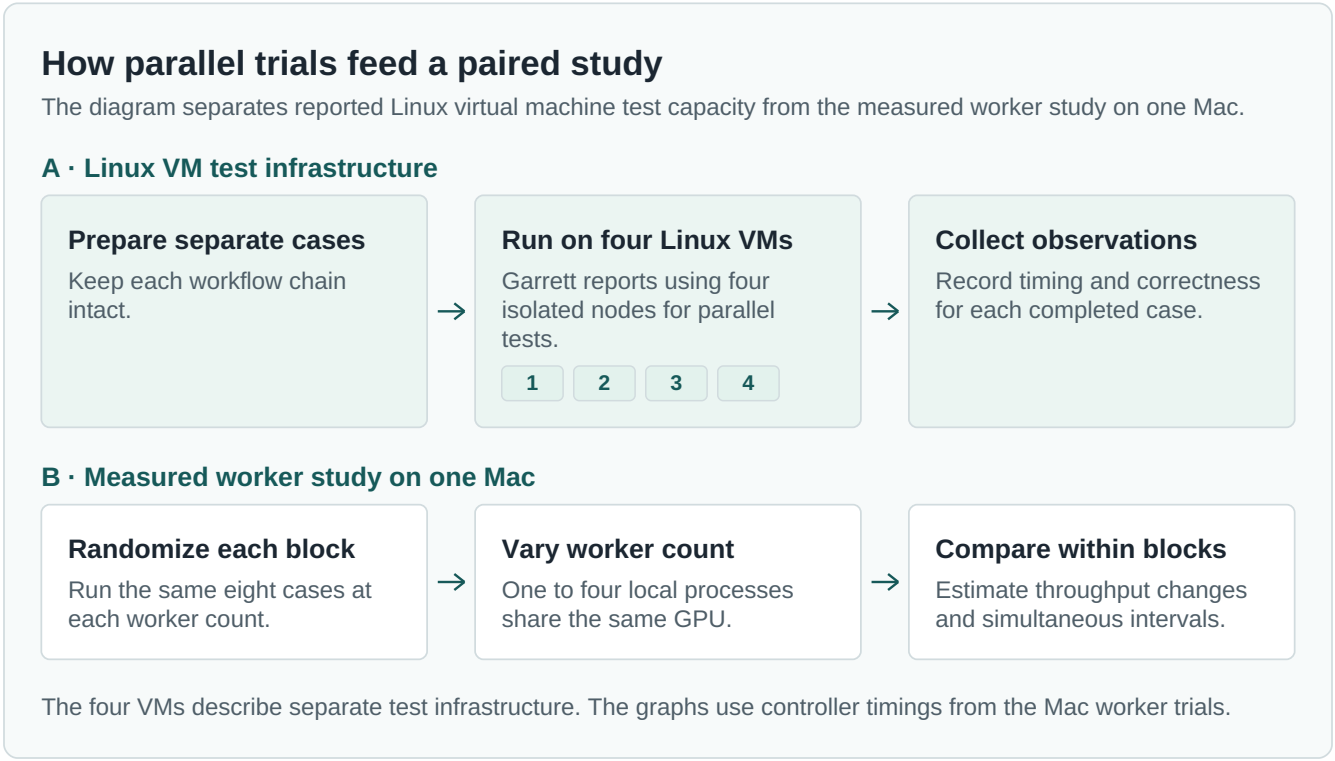


Figure 1. Parallel test capacity and statistical design serve different purposes. The Linux VM arrangement describes test infrastructure. The measured experiment ran every worker count within each randomized block on one Mac.

1. Research question

The n0is test harness evaluates a chain of ETL requests. Each request uses the workflow state left by the previous one. A model proposes the action, the C++ evaluator executes the workflow, and the harness verifies

the result. We asked how many such chains the controller could complete per minute at different worker counts, and what that meant for the latency of a single chain.

The engineering team used four isolated Linux VMs for separate test cases. The archived throughput studies described here used a separate instrumented setup with local processes on one Apple M5 Max sharing one GPU. The figures assign measured effects to that process setup. The Linux VM diagram documents the test infrastructure reported by Garrett.

2. Experimental design

We froze eight previously exercised tuning cases and kept each complete chain on one worker. Each block ran four rounds, one each at one, two, three, and four workers. The harness randomized the order of the four settings. Every round ran the same eight cases with static assignments. The study completed 48 rounds across 12 blocks, covering 384 chains. All 384 passed the recorded checks and reconciled with the verified result receipts.

The throughput clock covers controller time for the timed round, including native execution and independent grading. Throughput equals eight completed chains divided by elapsed round minutes. Individual latency uses the elapsed timer for a full chain and includes its model calls and native execution. These clocks answer different operational questions.

The analysis pairs rounds within the same block, giving 12 observations per worker-count contrast [1]. The reported throughput changes are ratios of geometric means. The analysis used paired sign-randomization tests with a Holm correction across six contrasts. It used Bonferroni-adjusted paired t intervals on log throughput ratios to give simultaneous 95% coverage for the six comparisons under the model assumptions. The figures show the three comparisons involving four workers.

3. Results

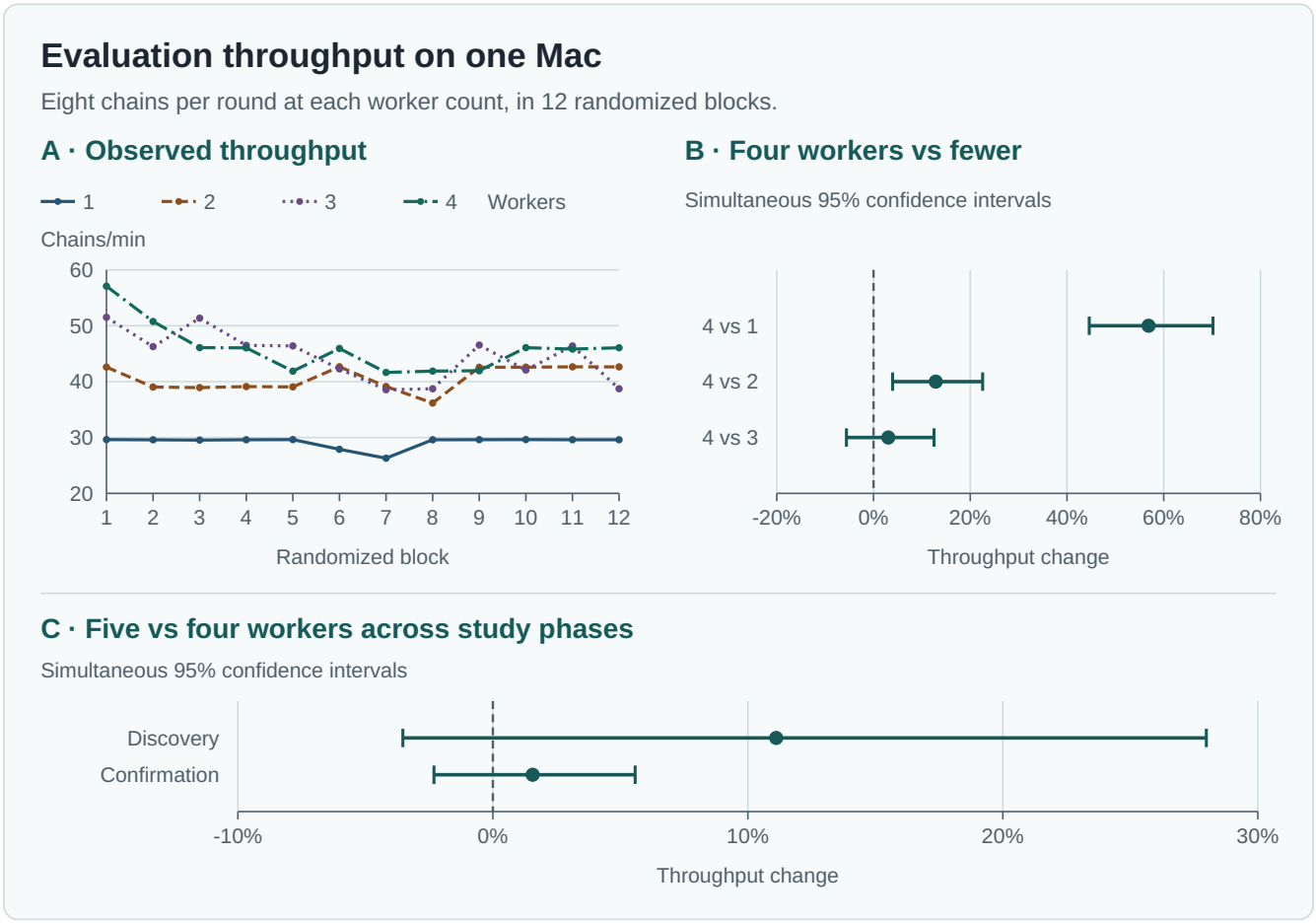


Figure 2. The trace shows one rate for each of 48 timed rounds. The interval plots report simultaneous 95% confidence intervals. The zero line marks equal throughput. Discovery and confirmation were separate experiments.

Local workers	Chains passed	Geometric mean chains/min	Mean chain latency (s)
1	96/96	29.16	1.84
2	96/96	40.53	2.44
3	96/96	44.39	3.13
4	96/96	45.75	3.76

Table 1. Throughput summarizes the timed rounds. Individual latency is the arithmetic mean of the chain clocks.

Four workers increased throughput by 56.87% over one (simultaneous 95% interval 44.60% to 70.18%) and by 12.87% over two (3.94% to 22.56%). The comparison with three gave 3.06%, with an interval from -5.60% to 12.51%. Mean individual chain time rose from 1.84 seconds at one worker to 3.76 seconds at four as the chains overlapped. Higher batch throughput made repeat testing quicker. The block design and paired estimates defined the uncertainty.

4. Independent confirmation

A separate discovery experiment compared four to eight workers across 12 blocks, 60 rounds, and 480 chains. Five had the highest observed throughput, with an estimated gain of 11.11% over four and a simultaneous 95% interval from -3.53% to 27.98%. A new 12-block experiment compared four, five, and six workers in 36 rounds and 288 chains. Five's estimated gain over four fell to 1.56%, with an interval from -2.31% to 5.58%. The team retained four as the working setting for short batches.

5. Scope and replication

The eight cases were fixed tuning workloads, so the 384 passing chains describe execution integrity for those cases. The paired blocks are the analysis units. Each study measures a particular model, machine, and controller boundary. A replication on the four Linux VMs would preserve the case assignments, record per-VM resources and timings, and repeat the randomized blocks under pinned configurations. Report controller and per-chain timing separately so an operator can choose between batch throughput and interactive latency [2].

Further reading

- Comparing Swift with the n0is C++ engine
- First and warmed CSV loads in Zig and n0is C++
- Scheduling a Local LLM Under a Live Microphone from Dev3lop

References

1. Kalibera, T., and Jones, R. Quantifying Performance Changes with Effect Size Confidence Intervals. arXiv, 2020.
2. Raasveldt, M., Holanda, P., Gubner, T., and Mühleisen, H. Fair Benchmarking Considered Difficult: Common Pitfalls in Database Performance Testing. DBTest, 2018.