

LithosAnanke Multiuser/Multitasking Design of Experiments

Concurrent Worker VM Load Across Three Instruction-Set Architectures

R. A. James (Captain Bob)

StarForth Project robert.allan.james@gmail.com

Generated automatically by `scripts/build_multiuser_doe_report.sh`
(Patent Pending — Provisional Filed December 2025)

Abstract

This report analyses the multiuser/multitasking Design of Experiments (FABRIC-3.md §XXXIII/§XXXV): a 6-cell factorial over concurrency level (2, 4, or 8 simultaneously live worker VMs, each birthed and touched twice by the Mama VM Hera) crossed with workload mode (uniform – every worker runs the same workload; heterogeneous – worker n runs workload $n \bmod 10$), run per instruction-set architecture with a fixed seed. Trial outcomes – concurrency, mode, fail count, and trial-end fleet physics (cumulative preemptive switch count, fleet heat conservation) – persist to a dedicated block-backed region on `disk/artemis.img` (DOE-PERSIST-TRIAL, §XXXV.10) rather than depending on a live serial-log tail, so this report can be rebuilt at any point during a multi-day campaign, including while it is still running, from whatever trials have completed so far.

1 Current campaign status

This report reflects **2** completed trial(s) persisted via DOE-PERSIST-TRIAL as of the last extraction, spanning ISA(s) `amd64`. Overall fail rate across all trials so far is **0.0%**; **100.0%** of trials read `fleet_conserved==1` at trial end.

- 2 workers, uniform: 1 trials, fail rate 0.0%, mean fleet K = 1.0000
- 8 workers, heterogeneous: 1 trials, fail rate 0.0%, mean fleet K = 1.0000

2 Fail rate by concurrency and workload mode



Figure 1: Fraction of trials with `fail_count` > 0, by concurrency level (2/4/8 workers) and workload mode. Labels give (failed/total) trials per cell.

3 Preemptive switch load

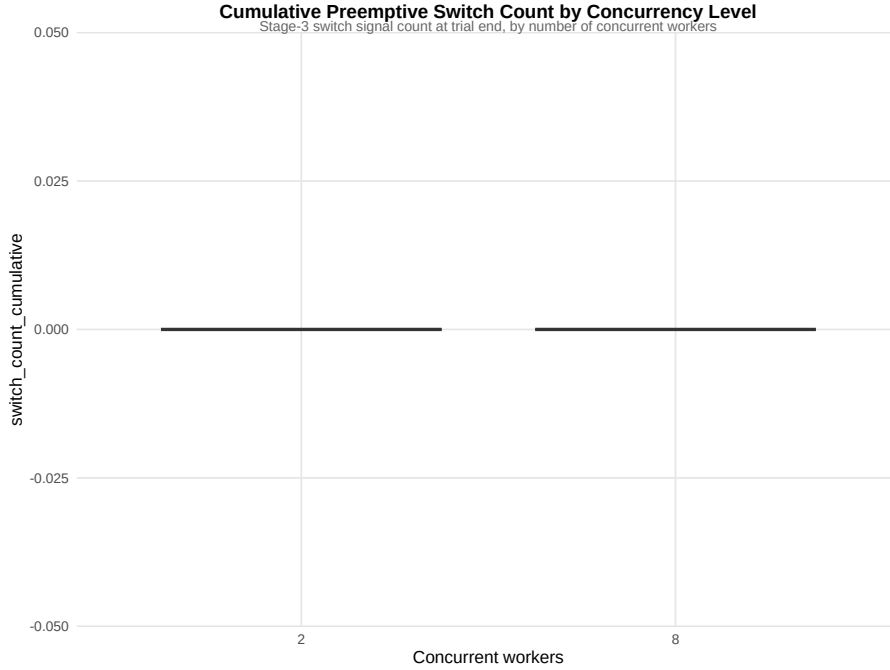


Figure 2: Distribution of cumulative Stage-3 preemptive switch count at trial end, by concurrency level. Higher concurrency is expected to drive more switch events per trial.

4 Fleet heat conservation across the trial sequence

5 Methodology notes

- **Persistence, not a serial-log dependency.** Earlier attempts at this campaign relied entirely on a live QEMU serial-log tail and were lost when the process was interrupted (FABRIC-3.md §XXXV.10). Every number in this report comes from `disk/artemis.img`'s own on-disk DoE region, read directly by `scripts/extract_doe_region.py` with no live QEMU process required.
- **Rebuildable mid-campaign.** This report's build script (`scripts/build_multiuser_doe_report.sh`) can be re-run at any point – including while a multi-day campaign is still executing – and will reflect however many trials have completed at that moment.
- **Trial-summary, not per-tick.** Unlike the per-tick HB-ON serial stream (which remains the authoritative live instrumentation during a run), the persisted region stores one summary record per completed trial (§XXXV.4) – sized to comfortably fit the device budget for the full campaign, not a raw tick mirror.

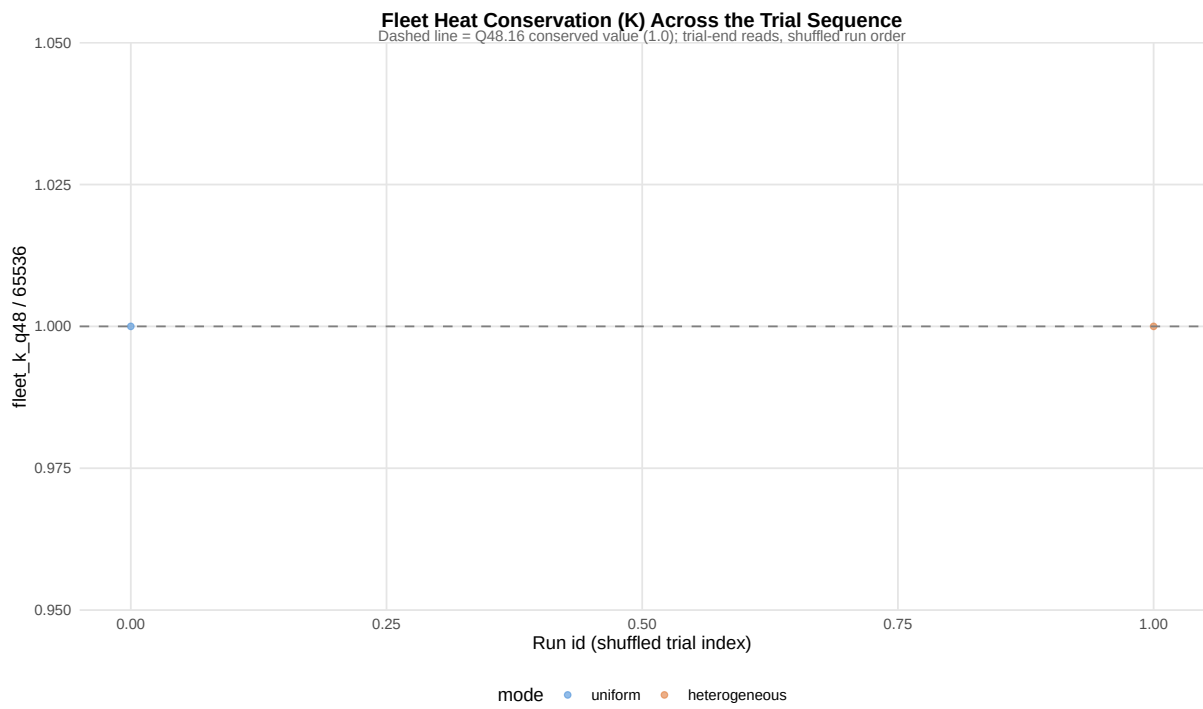


Figure 3: Trial-end fleet heat (`fleet_k_q48`, decoded from Q48.16) across the shuffled trial run order. The dashed line marks the conserved value (1.0); persistent drift away from it across the sequence would indicate a real conservation violation rather than per-trial noise.