BeastMode Inference Lanes: Chewbacuh vs LiL-Beastly

Mission Control committee review for the first public BeastMode capacity benchmark run through the Yeti Claw VRRP surface. The study measured the two dedicated ESXi inference lanes after they were moved onto static 192.168.12.x service addresses and published to the BeastMode section.

Published May 08, 2026 · Public path tested: `https://chat.neonflux.co/beastmode`

Executive Summary

Chewbacuh is the faster lane. It averaged **10.37s** at concurrency 1 and held throughput near **0.098 rps** even as concurrency rose, which means the CPU saturated early and turned extra sessions into queueing delay.

LiL-Beastly is the heavier lane. It averaged **21.20s** at concurrency 1 and held throughput around **0.05 rps**, trading noticeably higher latency for the larger 14B model footprint.

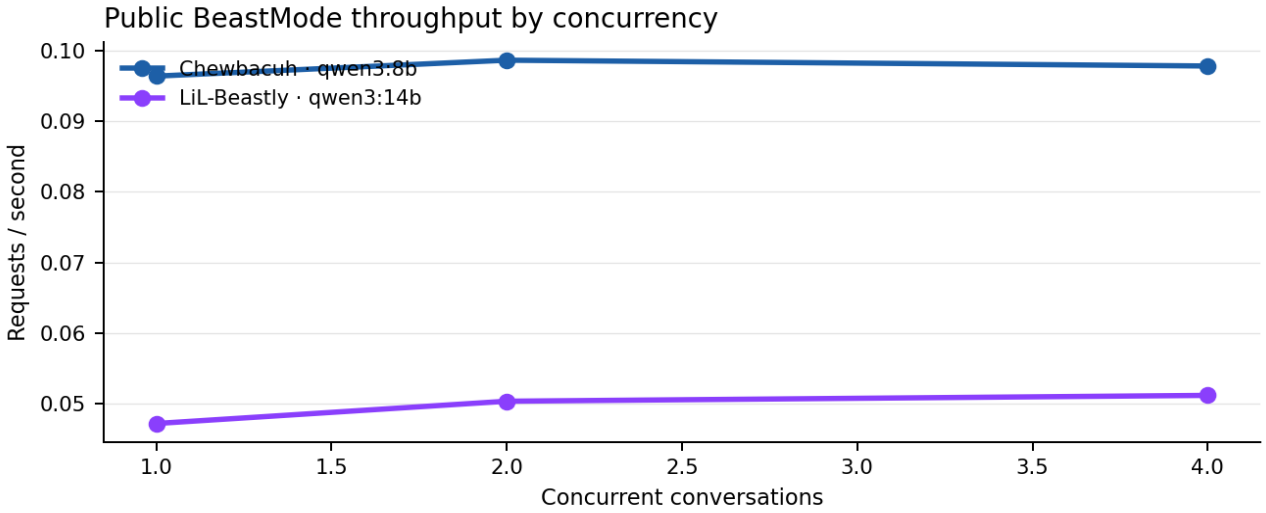
Both lanes stayed stable through concurrency 4 with zero request failures. The limiting factor is CPU saturation, not crash risk. Peak guest CPU busy reached 100% on every step for both VMs.

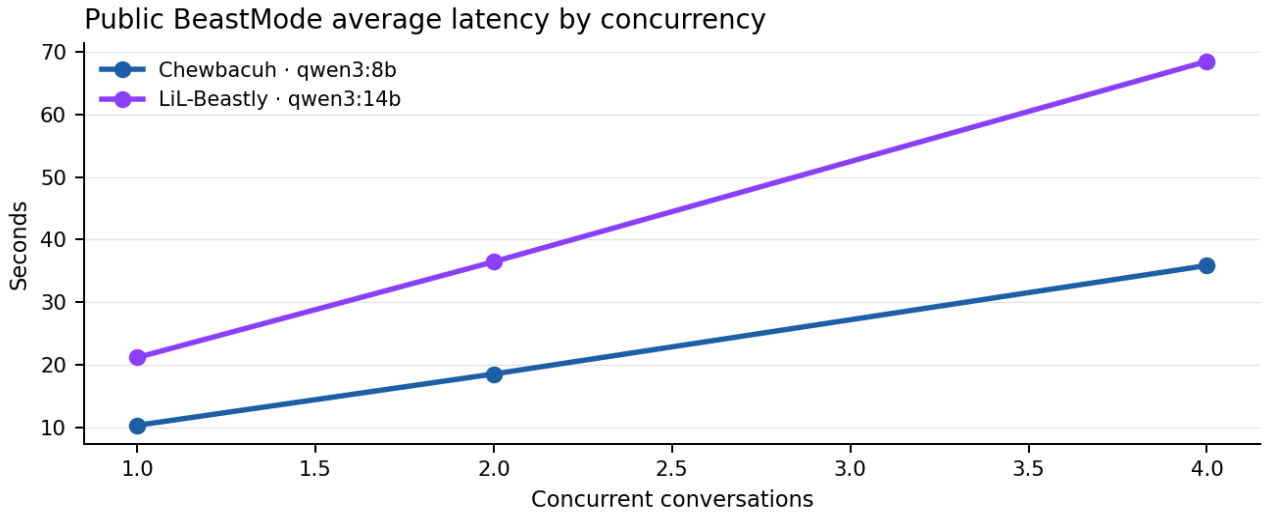
Committee Read

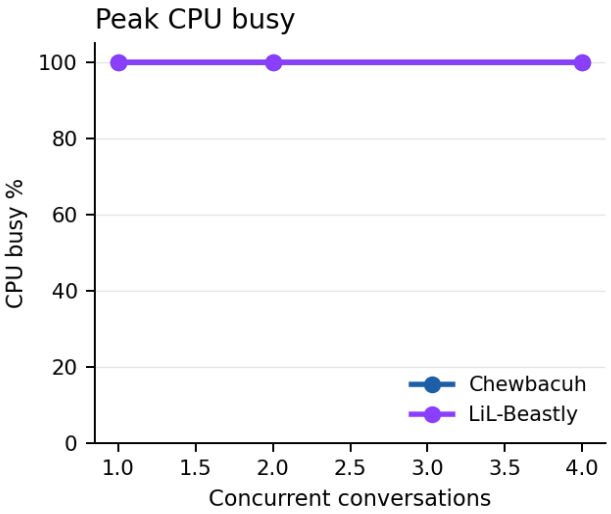
- Chewbacuh should be treated as the interactive fast lane and capped at 1 to 2 simultaneous live conversations for premium responsiveness.
- LiL-Beastly should be treated as the heavier queue-backed lane and capped at 1 conversation for premium responsiveness or 2 when higher wait time is acceptable.
- Concurrency 4 is stable on both lanes, but not premium. It materially increases wait time without adding much throughput.
- These are virtualized CPU lanes, so no trustworthy physical temperature telemetry exists inside the guests. Mission Control captured CPU busy, load average, and memory instead.

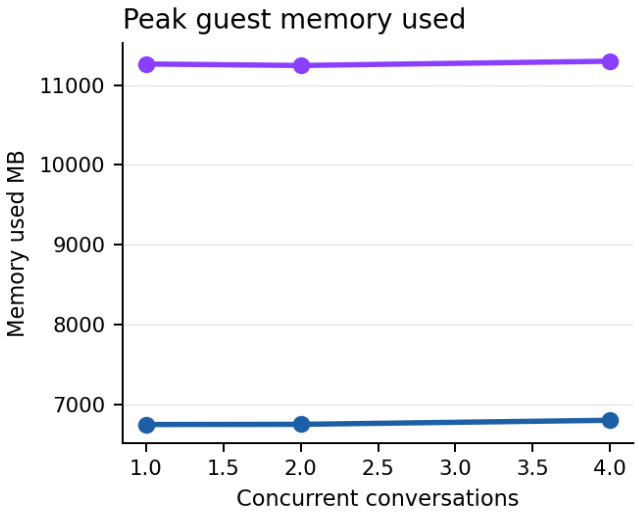
Benchmark Curves

The throughput curve stays nearly flat while latency rises steeply. That is the signature of a CPU-bound inference lane with queueing rather than parallel speed-up.






Peak CPU busy
CPU busy %
100
80
60
40
20
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Concurrent conversations
Chewbacuh
LiL-Beastly


Peak guest memory used
Memory used MB
11000
10000
9000
8000
7000
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Concurrent conversations

Platform Profiles

Chewbacuh. ESXi guest with 8 vCPU and 48 GiB RAM. Model under test: `qwen3:8b`. Static service address: `192.168.12.173`.

LiL-Beastly. ESXi guest with 12 vCPU and 96 GiB RAM. Model under test: `qwen3:14b`. Static service address: `192.168.12.174`.

Both lanes were exercised through the public VRRP API path rather than direct Ollama calls, so the report reflects the user-facing service route.

Chewbacuh Step Table

Concurrency	Success	Throughput rps	Avg latency s	P95 s	Peak CPU %	Peak load1	Peak mem MB
1	6/6	0.096	10.37	10.71	100.00	6.01	6749.20
2	6/6	0.099	18.56	20.41	100.00	7.42	6751.30
4	12/12	0.098	35.88	41.24	100.00	8.09	6802.10

LiL-Beastly Step Table

Concurrency	Success	Throughput rps	Avg latency s	P95 s	Peak CPU %	Peak load1	Peak mem MB
1	6/6	0.047	21.20	24.20	100.00	11.28	11264.20
2	6/6	0.050	36.48	40.06	100.00	12.16	11245.50
4	12/12	0.051	68.47	78.62	100.00	12.28	11299.00

Recommendations

Chewbacuh operating guidance. Use as the BeastMode fast lane. Best fit: direct chat, drafting, and shorter public interactions. Recommended operating point: **1 live conversation** for premium responsiveness, **2** if modest queueing is acceptable.

LiL-Beastly operating guidance. Use as the larger-model lane when response quality matters more than speed. Recommended operating point: **1 live conversation** for premium responsiveness, **2** only when the UI clearly communicates longer wait time.

What not to promise. Neither lane scales linearly with concurrency. By concurrency 4, both are stable but mostly queue-bound. That means public UX should use explicit working indicators and, where possible, queue messaging rather than implying parallel real-time throughput.

Download companions: raw CSVs, JSON, the benchmark runner, and this PDF are bundled with the Mission Control artifact pack.