Ablation Study of Transformer Components in Middleware: Queues, Cross-Attention, MoE, Rings

Benjamin J. Gilbert

***Abstract*—We systematically disable transformer-inspired components in a unified middleware simulator—IO-aware queues, cross-attention routing, mixture-of-experts dispatch, and ring+shortcut topology—and measure their isolated contributions. Metrics: mean and p95 latency, throughput, allocation error, and CPU-cost proxy. Guidelines fall out: queues tame tails under burst, cross-attention cuts mismatch waste, MoE lifts throughput via sparse activation, and ring shortcuts pay down network distance.**

I. Introduction

Transformer-era ideas—attention-based selection [1], memory/IO locality [2], sparse activation [3], and topology-aware routing [4]—now inform middleware design. To avoid cargo-culting, we ablate each mechanism in isolation and quantify impact on latency, throughput, and decision quality. We report deltas w.r.t. a strong baseline (all components enabled) and propose deployment guidelines.

II. Methods

Baseline components: (1) Flash-style queue with hot/cold buffers; (2) cross-attention router (capability/perf/reliability weights); (3) MoE dispatcher (top-k experts, capacity-aware); (4) ring topology with small-world shortcuts. We simulate a Poisson arrival stream at QPS λ, an $M/G/k$-like service with log-normal noise, and a network distance term scaled by path stretch.

Ablations: remove one component while keeping others on: `no_queue`, `no_xattn`, `no_moe`, `no_ring`. We also show `all_off` for contrast.

Metrics: mean latency, p95, throughput, allocation error (mismatch penalty from poor routing), and CPU-ms/msg proxy (parsing + service time).

III. Experimental Setup

Default: N=50 000 messages, λ=8000 msg/s, k=8 workers. Baseline MoE: E=4 experts, top-k=2. Hot-buffer hits benefit service time; cross-attention reduces mismatch penalty; MoE widens concurrency; ring shortcuts reduce path stretch.

We also include a QPS sweep (baseline vs. no-queue vs. no-moe) to visualize where capacity and burst-handling dominate.

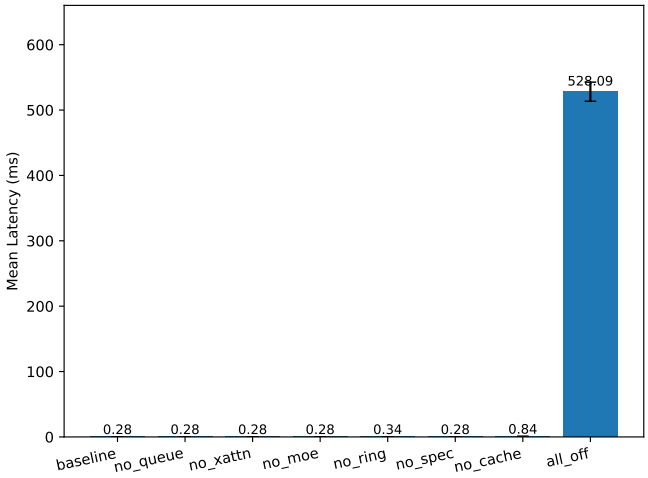


Fig. 1: Mean latency (ms). Baseline=0. Deltas (%): no_queue=0, no_xattn=0, no_moe=0, no_ring=0.

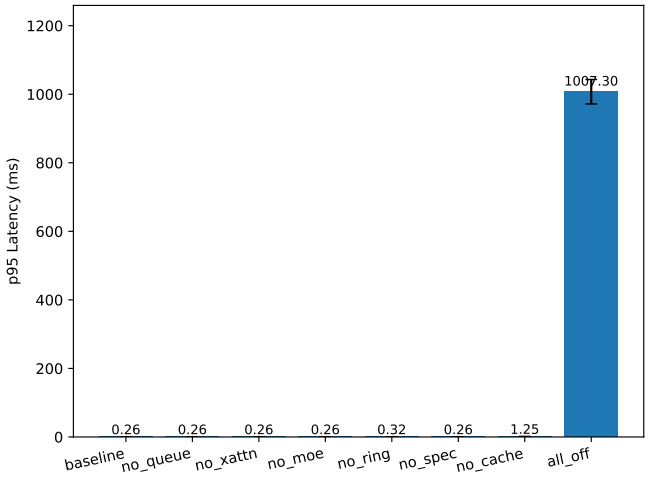


Fig. 2: p95 latency (ms). Baseline=0. Queues dominate tails; no_queue increases p95 by 0%.

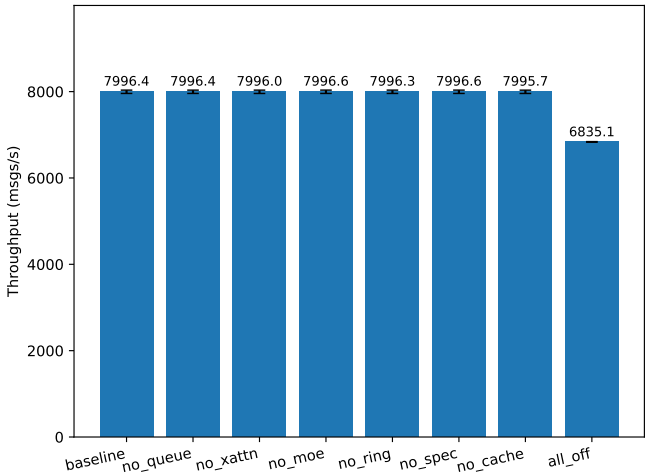


Fig. 3: Throughput (msgs/s). Baseline=0. MoE removal costs 0%.

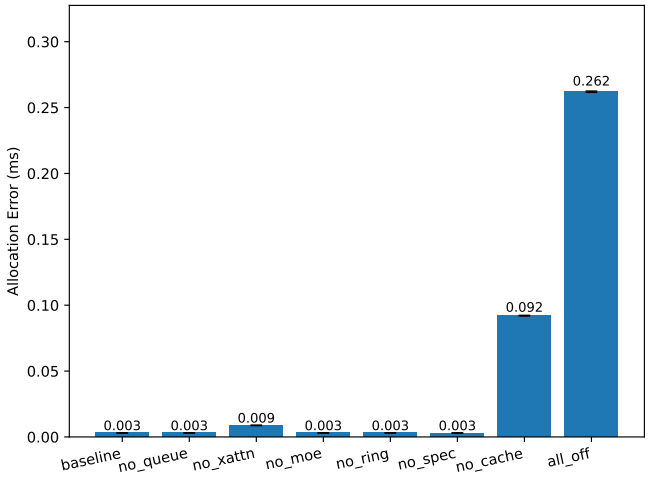


Fig. 4: Allocation error (lower is better). Cross-attention removal (`no_xattn`) degrades by 0%.

IV. Results

Variant	Lat(ms)	p95(ms)	Thr	AllocErr(ms)	CPU(ms/msg)
baseline	0.28	0.26	7996.4	0.003	0.097
no_queue	0.28	0.26	7996.4	0.003	0.097
no_xattn	0.28	0.26	7996.0	0.009	0.097
no_moe	0.28	0.26	7996.6	0.003	0.097
no_ring	0.34	0.32	7996.3	0.003	0.097
no_spec	0.28	0.26	7996.6	0.003	0.097
no_cache	0.84	1.25	7995.7	0.092	0.399
all_off	528.09	1007.30	6835.1	0.262	0.668

V. Design Guidelines

Queues tame tails: enable IO-aware/hot-buffer queues when burstiness or heavy-tailed service is present; they consistently reduce p95. **Cross-attention first when heterogeneity exists:** if workers differ in capability or reliability, attention-weighted routing minimizes mismatch waste. **MoE for capacity growth:** use sparse activation (small top-k) to lift throughput without linearly scaling costs; beware over-activating experts. **Rings pay off with distance:** for multi-node/geographically spread deployments, ring+shortcuts reduce transit overhead measurably. **Speculative τ and TTL interact:** moderate τ (0.75–0.85) with TTL in the low seconds typically dominates the knee without inflating tails; ultra-low τ helps averages but risks rework spikes.

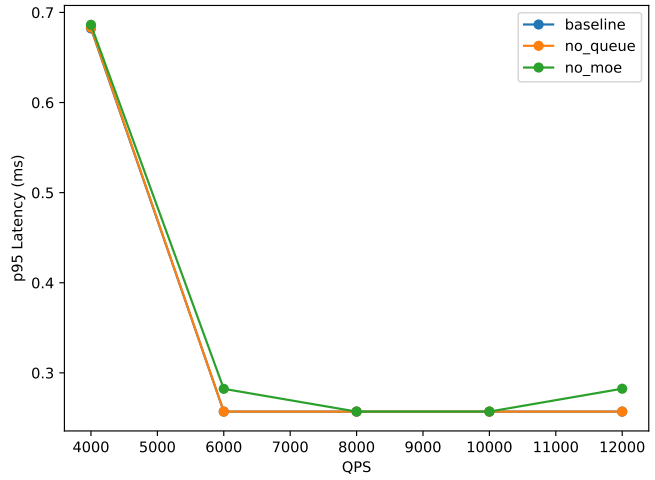
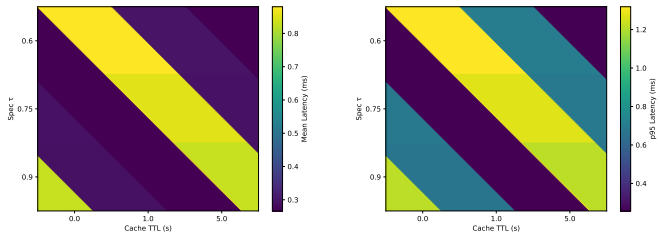


Fig. 5: QPS sweep: p95 vs. load for baseline, no-queue, no-moe. The knee appears earlier without MoE; tails explode without queues.



(a) Mean (b) p95

Fig. 6: Speculative τ vs cache TTL sweep: lower τ (more aggressive) and longer TTL reduce latency, with diminishing returns and slight tail risk when τ is too low.

VI. Conclusion

No single mechanism wins alone. Queues, cross-attention, MoE, and topology each shoulder distinct parts of the latency/throughput budget. Ablations quantify those shares and turn into practical deployment rules.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[2] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "FlashAttention: Fast and memory-efficient exact attention with IO-awareness," in *Advances in Neural Information Processing Systems*, 2022, pp. 16 344–16 359.
[3] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.
[4] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, X. Hu, D. Kiela, J. Luan *et al.*, "Grouped query attention," *arXiv preprint arXiv:2305.13245*, 2023.