

An Optical KV Cache Network for Disaggregated LLM Serving

Seongwon Yoon and Shimeng Yu

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA
gabrielyoon@gatech.edu

Abstract—In disaggregated LLM serving, each request’s KV cache moves from a prefill GPU to a decode GPU before the response begins, and a coding-agent turn often moves the KV cache of its whole long context. Behind an oversubscribed packet fabric, this transfer adds to the time to first token (TTFT). An optical circuit switch (OCS) can give every transfer a full-rate circuit for less than an oversubscribed packet fabric costs, but a mechanical OCS needs tens of milliseconds to reconfigure. We hide this delay inside the prefill. A reservation scheduler commits each circuit while the prefill is still computing, so the circuit is ready when the KV cache is, and each GPU reaches the OCS through its own wavelength-multiplexed (WDM) optics on one fiber pair. In simulations of GPU clusters on real serving traces with prefill times measured on H100 GPUs, this optical KV cache network matches 400G packet fabrics with adaptive routing at 0.55 to 0.78 times the cost of a 2:1 fabric and tolerates the 50 ms reconfiguration delay of a 384-port mechanical OCS.

I. INTRODUCTION

Prefill/decode (P/D) disaggregation runs the two phases of LLM serving on separate GPU pools [1], [2], [3]. Every request’s KV cache then crosses the network and adds to TTFT unless the transfer ends within the prefill. In our analysis of one P/D application in Alibaba’s production trace [4], the hourly mean KV transfer time is a median 1.10 times the prefill time over one week. Deployed P/D systems choose the decode instance by load [3], [5], [6], so an agent turn [7] that appends a tool result to a long context often moves the KV cache of the whole context, while its prefill computes only the new tokens. A non-blocking 400G fabric mostly hides this move, but operators oversubscribe the upper tiers [8], [9], [10]. At half of 400G per GPU, the median migrated turn of Llama-3.1-70B on H100 GPUs adds 67 ms to TTFT.

A non-blocking fabric costs 1.27 to 1.31 times a 2:1 fabric, and packet schedulers [8], [11] reduce contention on the same links. An OCS port draws far less power than a packet switch port and passes light without transceivers [12], [13], so a circuit per transfer can give every GPU its full link rate for less than an oversubscribed fabric costs. However, a mechanical OCS needs 10 to 50 ms to reconfigure [12], [14], [15]. We hide this delay inside the prefill. Earlier OCS fabrics reconfigure per training job, phase or workload mode [12], [13], [16], [17], [18], [19], while our circuits follow single KV transfers. Our contributions are (1) an optical KV cache network with GPU-side WDM optics and the requirements it sets on them, namely one fiber pair per 400G link, up to 50 ms of reconfiguration and 2 dB of link margin (Section III-A, Table I), (2) a reservation scheduler that commits the prefix circuit at prefill start and the

new-KV circuit at the last prefill chunk and reuses standing circuits (Section III-B), and (3) simulations of H100 clusters on real serving traces with measured prefill times (Section IV).

II. BACKGROUND

A. Prefill, decode, and KV cache transfer

Prefill reads the whole prompt at once, and decode generates one token at a time, both through the key and value vectors that attention stores per token and layer, the KV cache. For Llama-3.1-70B in BF16 it takes 320 KiB per token [20], so a 64K-token context occupies about 21 GB. Prefill is compute bound and decode is memory bound, so P/D disaggregation runs them on separate pools [1], [2], as in Mooncake, NVIDIA Dynamo and DeepSeek [3], [5], [6]. Mooncake streams the KV cache layer by layer during prefill, which hides the transfer while it ends within the prefill. The vLLM NIXL connector pulls it after prefill [21], which adds the whole transfer to TTFT, because decode emits the leading token.

B. Multi-turn agent sessions and KV cache migration

Coding agents append a short input, such as a tool result, to a long shared context every turn. In a trace of 4,300 coding-agent sessions from 43 developers [7], the steps that fit the 128K window of Llama-3.1-70B have a median context of 66.7K tokens and a median new input of 1.1K tokens, against 88.8K and 1.6K for main-agent turns in AgentX Claude Code sessions [22] and a 68K-token median prompt in GitHub Copilot [23]. With prefix caching, the prefill side keeps earlier turns’ KV cache and computes only the new tokens [24], [25], [26]. A turn migrates when the chosen decode instance does not hold the prefix, so its full KV cache crosses the network before decode can respond. Mooncake and DeepSeek send the full KV cache to a decode instance chosen by load every turn, and Dynamo ignores prefix reuse when routing decode [3], [5], [6]. In llm-d runs on coding-agent traces, about half of the decode-side KV cache arrived from prefill [27]. In Alibaba’s production P/D telemetry [4], the least-utilized decode instance of a group changes in 42% of consecutive 5-minute windows. Session affinity keeps a session on one decode instance, but 7.6% of turns still miss decode memory in our modeled replay, and a KV store still sends the KV cache over the fabric [5].

C. Packet and circuit switching in the datacenter

A GPU has three network paths. NVLink joins the GPUs of a server, a 400G RDMA NIC per GPU joins GPUs across

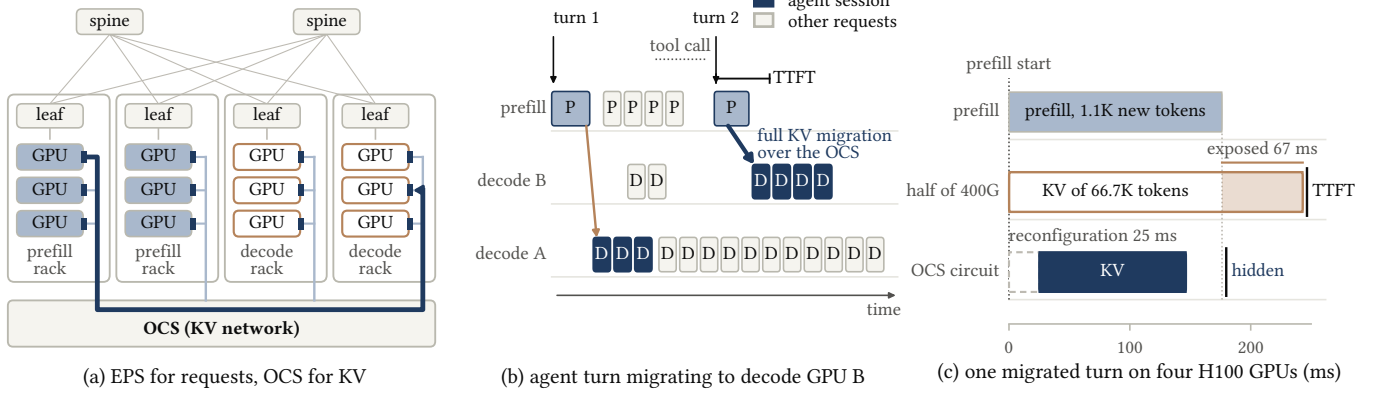


Fig. 1. Optical KV cache network. (a) Requests, output tokens and control stay on the host NICs and their leaf-spine EPS, while the RDMA NIC of every GPU drives WDM optics into an OCS instead of a packet leaf. (b) After a tool call, decode GPU A is busy with other requests, so turn 2 lands on decode GPU B and its full KV cache crosses the OCS on a circuit committed during prefill. (c) Median migrated turn of Llama-3.1-70B on four H100 GPUs, modeled with the measured prefill time. One circuit committed at prefill start reconfigures in 25 ms and ends inside the 176 ms prefill.

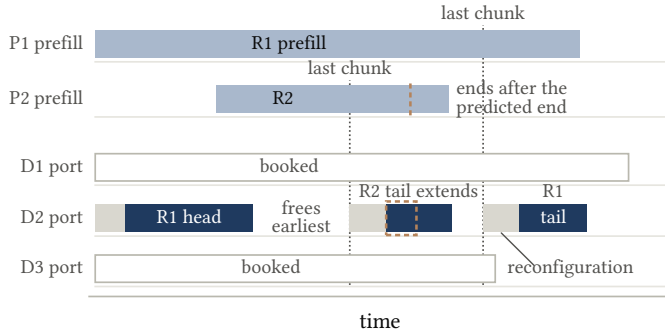


Fig. 2. Reservation calendar with last-chunk commit. R1 commits its head circuit at prefill start on the decode port that frees earliest (D2) and its tail at the last chunk, so the reconfiguration delay (grey) overlaps that chunk. The late tail of R2 extends into free port time.

servers [9], [10], and host NICs carry requests, storage and management [9]. Both NIC paths cross leaf and spine fat trees of electrical packet switches (EPS), whose upper tiers are often oversubscribed [8], [9], [10]. An EPS spreads flows by hashing (ECMP), which lets large transfers collide [28], or by adaptive routing [29] under RDMA congestion control [30]. An OCS steers light from an input fiber to an output fiber without converting it to an electrical signal [31], [32], [33], as in Google’s TPU and datacenter networks [18], [34], [35]. It does not inspect packets, so one port serves any line rate at 0.39 W against 39.4 W for a 400G packet switch port [13], and at \$520 against \$583 to \$1,090 [12], [13]. In exchange, a circuit joins one pair of ports at a time. Mechanical switches, steered by MEMS mirrors or piezoelectric actuators, take up to 25 ms to change it with 192 ports and 50 ms with 384 ports [12], [14], [15], and silicon photonic switches about a microsecond [36].

III. OPTICAL KV CACHE NETWORK

A. Architecture

Requests, output tokens and control stay on the host NICs and their leaf-spine EPS, while the 400G RDMA NIC of each

Algorithm 1 Reservation with last-chunk commit per request

- Admission**
- 1: $\hat{T} \leftarrow$ prefill time predicted by the cost model
 - 2: $B_h, B_t \leftarrow$ KV bytes of the prefix and of the new tokens
 - 3: split into head B_h and tail B_t only if $(B_h + B_t)/R + \tau \leq 0.8 \hat{T}$
- Prefill start** at t_0
- 4: $d \leftarrow$ decode port free earliest, preferring a standing circuit
 - 5: $\tau_h \leftarrow 0$ if the circuit to d stands else τ
 - 6: book head of length $\tau_h + B_h/R$ in the earliest gap after t_0 free on both ports
- Last prefill chunk** at t_1
- 7: $\hat{E} \leftarrow$ prefill end re-predicted from observed progress
 - 8: book tail on d to end at $\hat{E} + T_\ell + 0.1(\hat{E} - t_1)$
- Prefill end** at E
- 9: if the tail circuit is not up then move the tail to the earliest free gap
 - 10: if a burst outlasts its window then extend it into free port time, displacing bookings without ready data, else cut it and re-book the rest
 - 11: keep the circuit up for reuse

GPU connects to an OCS instead of a packet leaf switch and carries the KV cache of every migrated turn (Fig. 1(a), (b)). Each prefill and decode GPU drives one 400G link of plug-gable or co-packaged WDM optics [37] into the OCS, 45 GB/s at the 90% efficiency we assume for every fabric. A fiber OCS switches fibers, not wavelengths, so the GPU-side optics set its size. A 400GBASE-FR4 link carries four wavelengths on one fiber pair, while a parallel-fiber 400GBASE-DR4 link occupies four pairs [38] and needs four times the switch ports. Eight 384-port planes, each joining one GPU position of every 8-GPU node, reach 3,072 GPUs in one stage [15], so KV moves between equal GPU positions.

B. Reservation scheduler

The controller keeps one calendar per instance, listing the booked windows of its OCS ports and the peer each port points to. A window holds the reconfiguration delay, skipped when the circuit already stands toward that peer, and the transfer. A booking takes the earliest gap free on both the prefill and the decode port, so windows never overlap. Algorithm 1 lists the four events of a request (Fig. 2), with R the circuit rate, τ the

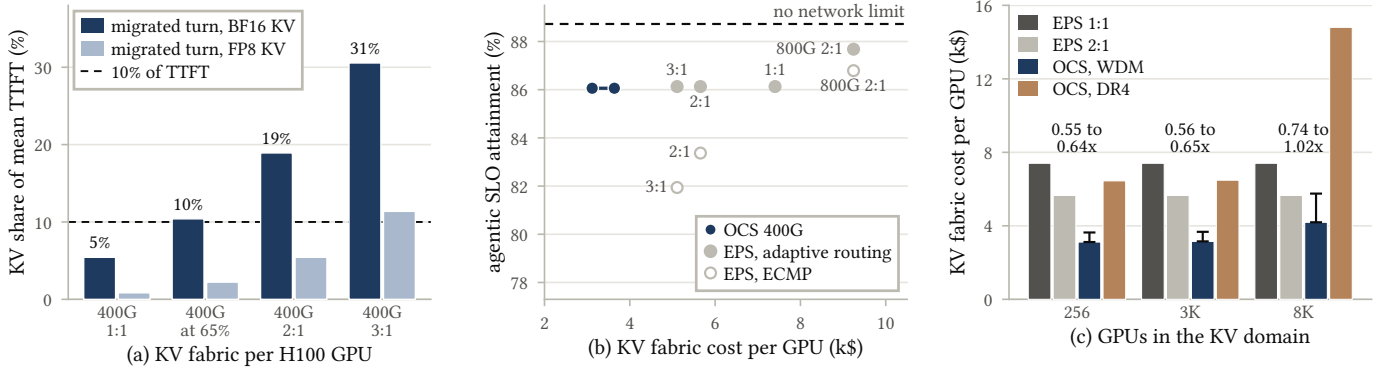


Fig. 3. (a) KV share of mean TTFT with each GPU at its share of 400G, Llama-3.1-70B on four H100 GPUs (modeled, layer-wise), chat turns below 0.1%. (b) Agentic SLO attainment against cost per GPU at 256 H100 GPUs (simulated, pull), the OCS at one or two ports per fiber pair. (c) KV fabric cost per GPU at 400G with the prices of [12]. Whiskers reach two ports per fiber pair, ratios to EPS 2:1 above each group, three stages at 8K and DR4 on eight ports.

reconfiguration delay and T_ℓ one prefill layer. The head of each transfer carries the prefix KV, which the prefill GPU already holds and only reads, so it does not depend on the tail, which carries the KV of the new tokens as their layers finish. We commit the head at prefill start, when its start time is exact, and the tail when the last prefill chunk of up to 2,048 tokens starts, which leaves only the error of the remaining prefill. A booking at admission would instead absorb the error of every prefill queued ahead. Preferring a standing circuit (line 4) makes the decode choice circuit-aware and favors reuse.

IV. EVALUATION

A. Setup

Design space. A request's KV demand per GPU D is the KV bytes one GPU sends over the prefill time, and S is the effective KV bandwidth per GPU, so layer-wise streaming exposes $\max(0, 1 - S/D)$ of TTFT. Prefill times come from a cost model fitted to 33 vLLM prefill runs of Llama-3.1-70B that we measured on four H100 GPUs (median error 4.6%), without scaling. We use H100 GPUs because they are the most advanced GPU system available to us, and the method applies in principle to newer GPUs by refitting the cost model.

Simulation. A flow-level event-driven simulator, driven by request traces as in prior P/D studies [1], [2], [11], models 128 and 256 H100 GPUs as slices of a larger fabric, with 32-GPU racks and prefill and decode in separate racks. Each decode instance is a KV sink without decode compute, so the results cover the KV fabric and TTFT. A TraceLab tenant [7], whose decode instances keep no session KV between turns as in Mooncake and DeepSeek, runs beside Azure 2024 chat and code [45], [46] and Mooncake [3] tenants, all on the measured Llama-3.1-70B configuration on four H100 GPUs. The base point gives TraceLab 25% of the GPUs at 60% prefill load, and the grid adds a 50% share and 75% load, in layer-wise and pull transfer. Results average five paired 330s trace windows, with prediction errors drawn from the measured runs. SLO attainment is the share of requests with TTFT within five times the tenant's median unloaded prefill time. Without the OCS, the KV crosses a two-tier EPS at 400G or 800G per GPU

TABLE I
OPTICAL KV CACHE NETWORK REQUIREMENTS AND LINK BUDGET

Requirement	Value	
KV bandwidth per GPU	33 GB/s for 10% of mean TTFT	
Reconfiguration delay	50 ms, every tenant within 5% at 60% load	
OCS unit	384 ports, 50 ms, 8 planes reach 3,072 GPUs	
GPU-side optics	WDM, one fiber pair per 400G link at 45 GB/s	
Optics	DR4 (parallel)	WDM (FR4)
OCS ports per link	4 or 8	1 or 2
Cost over 2:1 EPS	0.77 to 1.36	0.55 to 0.78
Break-even port price	\$305 to \$842	\$1,050 to \$3,058
Link budget	DR4	WDM
<i>one 400G link over 500 m of fiber, transceiver to OCS to transceiver</i>		
Launch OMA (dBm)	-0.8 to +4.2	-0.3 to +3.7
Rx sensitivity (dBm)	max(-3.9, SECQ-5.3)	max(-4.6, SECQ-6.0)
Mux and demux (dB)	none	2.5 typ, 3.3 max each
OCS (dB)	1.5 typ, 2.7 max	
Four connectors (dB)	2.0 typ, 3.0 max	
Fiber at 0.5 dB/km (dB)	0.25	
Path loss (dB)	3.75 typ, 5.95 worst	
Loss allocation (dB)	3	4
Margin at TDECQ 3.4 dB	2.35 typ, 0.15 dB worst	2.55 typ, 0.35 dB worst
TDECQ for 2 dB margin	≤ 1.5 dB	≤ 1.7 dB
TDECQ for 2 dB, spliced	≤ 2.4 dB	≤ 2.6 dB

Sources are the exposure model, Fig. 4(c), the cost model [12], [13], data sheets [15], [39], [40], [41] and specifications [38], [42], [43], [44]. Margins use the maximum launch and the 1550nm OCS loss, with mux and demux in the module.

and at 1:1, 2:1 or 3:1 (3:1 as in MixNet [12], simulated as 2.91:1 with 11 uplinks per 32 GPUs). The EPS uses ECMP, adaptive routing with fair sharing as in RDMA fabrics [29], [30], or adaptive routing with global SRPT as a strong packet reference [47], [48], and a network without limits gives the ceiling. The OCS runs last-chunk commit at a 25 ms delay, and 50 ms changes its SLO attainment by at most 0.15 percentage points at 60% load. A baseline without reservation holds a circuit as soon as any KV is ready, with the decode chosen by load.

Cost. Both designs keep the same host NIC network, so for new clusters that serve each model within one node, as here, we compare only the network behind each GPU's 400G RDMA NIC. Switch ports, NICs and 400G modules use published prices [12], [13], and an FR4 module costs 1.47 times a 400G module as in reseller lists. The EPS uses the fewest tiers of 51.2T switches, and the OCS one or two ports per fiber pair [13], [39] and a 100G control network.

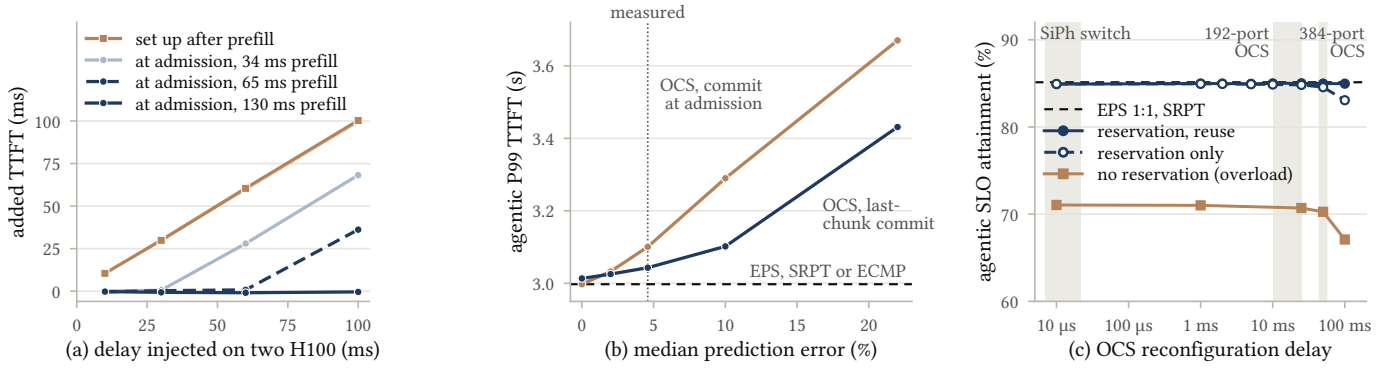


Fig. 4. (a) TTFT added by a circuit delay injected after prefill or at admission on two H100 GPUs. (b) Agentic P99 TTFT against median prediction error at 1 ms and (c) agentic SLO attainment against reconfiguration delay, simulated at the 128-GPU base point, layer-wise, where no network limit reaches 85%.

B. Performance and cost

Migrated turns of Llama-3.1-70B demand a median 28.6 GB/s per H100 GPU, which fits in a 400G link at 45 GB/s. A non-blocking fabric therefore exposes 5% of mean TTFT, or 10% at the 32.5 GB/s NCCL rate we measured over 400G InfiniBand, and half or a third of 400G per GPU exposes 19% and 31% (Fig. 3(a)). At these rates AgentX turns expose 12% and 20%, and FP8 KV cuts the TraceLab shares to 5% and 11%.

Under load at 128 GPUs, the OCS stays within 0.4 points of the agentic SLO attainment of EPS 1:1 with SRPT at every grid point and mode, with P99 TTFT within 1.8%. Adaptive routing with fair sharing stays within 0.5 points of the OCS even at 3:1, so the loss behind oversubscription comes from ECMP collisions of migrated turns. In pull transfer the OCS gains 3.4 to 6.6 points over 2:1 ECMP and 4.6 to 10.7 points over 3:1 ECMP. In layer-wise transfer it gains 0.2 to 3.2 points and hides 71 to 74% of migrated turns within 10% of their prefill time, against 50 to 58% for 2:1 and 43 to 53% for 3:1 ECMP. At 256 GPUs the pull gains over ECMP are 2.7 to 6.2 points (Fig. 3(b)). The gain follows the share of migrated turns. When half the turns migrate, as half the decode-side KV does in llm-d [27], the pull gains over 2:1 and 3:1 ECMP are 1.3 to 1.8 and 1.8 to 2.7 points, and under session affinity (7.6%) every fabric is within 0.6 points, where the OCS keeps its cost advantage.

At two sets of published prices [12], [13], with a NIC and a WDM pluggable per link, the OCS costs 0.55 to 0.78 times a 2:1 EPS and 0.61 to 0.85 times a 3:1 EPS from 256 to 3,072 GPUs, the reach of one OCS stage (Fig. 3(c) shows one price set). With parallel-fiber DR4 it costs 0.77 to 1.36 times a 2:1 EPS in one stage (Table I), so WDM optics pay off best.

C. What the optical layer must provide

Overlap. On our two-H100 testbed (Llama-3.1-8B over NVLink, Fig. 4(a)), an injected delay started at admission stays hidden while it is shorter than the prefill, and one started after prefill adds 0.95 to 1.12 times its length.

Prediction. With last-chunk commit, agentic P99 TTFT rises 1.5% at the measured 4.6% median error and 3.6% at 10%,

against 3.5% and 9.6% with admission commit (Fig. 4(b)), so P99 stays within 5% up to 11.6% error, against 5.9% with admission commit.

Reconfiguration delay. Without reservation, layer-wise transfers hold decode ports through every prefill. With the decode chosen by load, the Mooncake tenant then overloads in every trace window even with a 10 μs switch (agentic SLO attainment 71%), while reservation with the same decode choice holds 85% at 25 ms. Circuit-aware decode choice keeps the same decode for 92% of consecutive agentic requests, with the busiest decode at 1.07 times the mean, so with reuse the SLO attainment stays at 85% up to 100 ms (Fig. 4(c)). Without reuse, all tenants keep P99 and SLO within 5% of EPS 1:1 with SRPT up to 50 ms at 60% load, so a mechanical OCS suffices.

Link budget. The worst path through four connectors and a 384-port OCS at 2.7 dB [15] loses 5.95 dB, above the 3 and 4 dB channel allocations of DR4 and FR4, so 2 dB of margin needs a launch near the maximum and a TDECQ of at most 1.5 dB with DR4 and 1.7 dB with WDM, or spliced OCS fibers with typical loss for any compliant TDECQ at the maximum launch.

V. RELATED WORK

Optical datacenter networks reconfigure for aggregate traffic [31], [32] or training jobs [17], [49], or rotate through traffic-oblivious schedules [50], [51]. MoSE reallocates OCS ports per workload mode without modeling reconfiguration delay [16], xSwitch pairs a port-reduced EPS with an OCS for scale-up [52], a photonic CXL appliance replaces the P/D hop with a memory pool [53], and Opus names KV cache pipelines as a target it does not evaluate [13]. KV compression [54], [55] loses to raw transfer above 50 to 110 Gb/s [56].

VI. CONCLUSION

Committing each circuit during prefill hides the reconfiguration delay of a mechanical OCS, so an optical KV cache network matches 400G packet fabrics with adaptive routing at 0.55 to 0.78 times the cost of a 2:1 fabric. In pull transfer it raises agentic SLO attainment over 2:1 and 3:1 ECMP fabrics by 2.7 to 6.6 and 4.1 to 10.7 points.

REFERENCES

- [1] Y. Zhong *et al.*, “DistServe: disaggregating prefill and decoding for goodput-optimized large language model serving,” in *Proc. OSDI*, 2024.
- [2] P. Patel *et al.*, “Splitwise: efficient generative LLM inference using phase splitting,” in *Proc. ISCA*, 2024.
- [3] R. Qin *et al.*, “Mooncake: trading more storage for less computation – a KVCache-centric architecture for serving LLM chatbot,” in *Proc. FAST*, 2025.
- [4] Y. Zheng, L. Yang, Y. Yu, G. Yang, L. Zhang, and M. Yu, “Model-as-a-Service in the wild: characterizing production LLM inference clusters at scale,” in *Proc. ACM SIGOPS ATC*, 2026, trace cluster-trace-v2026-maas.
- [5] Y. Wu *et al.*, “DualPath: accelerating agentic LLM inference by harvesting disaggregated KV-cache storage I/O,” in *Proc. ACM SIGCOMM*, 2026.
- [6] NVIDIA, “Disaggregated serving,” Dynamo router documentation, v1.2.1, 2026.
- [7] K. Zhu *et al.*, “TraceLab: characterizing coding agent workloads for LLM serving,” arXiv:2606.30560, 2026.
- [8] H.-C. Lin, T.-W. Hsu, C.-E. Ho, and A. Saeed, “ARK: avoiding routing collisions for KV cache transfer in disaggregated LLM inference,” in *Proc. ACM SIGCOMM Workshop on Networks for AI Computing (NAIC)*, 2026, pp. 2067–2073.
- [9] K. Qian *et al.*, “Alibaba HPN: a data center network for large language model training,” in *Proc. ACM SIGCOMM*, 2024.
- [10] A. Gangidi *et al.*, “RDMA over Ethernet for distributed AI training at Meta scale,” in *Proc. ACM SIGCOMM*, 2024.
- [11] Y. Sun *et al.*, “Nuska: stage-aware communication scheduling for disaggregated LLM serving,” arXiv:2603.17456, 2026.
- [12] X. Liao *et al.*, “MixNet: a runtime reconfigurable optical-electrical fabric for distributed mixture-of-experts training,” in *Proc. ACM SIGCOMM*, 2025, pp. 554–574.
- [13] E. Ding, B. Lyu, B. Kataria, and R. Singh, “Opus: photonic rail-optimized fabric in ML datacenters,” in *Proc. ACM SIGCOMM*, 2026, pp. 922–939.
- [14] HUBER+SUHNER Polatis, “POLATIS Series 6000n network optical matrix switch,” data sheet, Rev. 6000n.022022.001, 2022.
- [15] HUBER+SUHNER Polatis, “POLATIS Series 7000n optical switch,” data sheet, Rev. 7000n.022022.001, 2022.
- [16] F. Yang, Y. Zhou, B. Wang, Z. Zhou, and J. Li, “MoSE: mode-switching expander for mixed LLM training and inference,” in *Proc. Asia Communications and Photonics Conf. (ACP)*, 2026, arXiv:2609.39138.
- [17] W. Wang *et al.*, “TopoOpt: co-optimizing network topology and parallelization strategy for distributed training jobs,” in *Proc. NSDI*, 2023.
- [18] N. P. Jouppi *et al.*, “TPU v4: an optically reconfigurable supercomputer for machine learning with hardware support for embeddings,” in *Proc. ISCA*, 2023.
- [19] X. Tan, Y. Feng, Y. Zhou, Y. Jiang, Y. Zhu, and H. Xu, “Dynamic compute and network orchestration for disaggregated RL,” in *Proc. ACM SIGCOMM*, 2026.
- [20] A. Grattafiori *et al.*, “The Llama 3 herd of models,” arXiv:2407.21783, 2024.
- [21] vLLM Team, “NixlConnector usage guide,” vLLM documentation, 2025.
- [22] SemiAnalysis, “AgentX v1.0 Claude Code traces,” dataset semianalysisai/cc-traces-weka-062126, Hugging Face, Jun. 2026.
- [23] B. Liu, H. Qiu, Í. Goiri, R. Fonseca, R. Bianchini, and E. Choukse, “Agentic coding in the wild: characterizing GitHub Copilot at production scale,” arXiv:2608.00101, 2026.
- [24] W. Kwon *et al.*, “Efficient memory management for large language model serving with PagedAttention,” in *Proc. SOSP*, 2023.
- [25] L. Zheng *et al.*, “SGLang: efficient execution of structured language model programs,” in *Proc. NeurIPS*, 2024.
- [26] V. Srivatsa *et al.*, “Preble: efficient distributed prompt scheduling for LLM serving,” in *Proc. ICLR*, 2025.
- [27] M. Ayoub, Y. Fama, E. Crncevic, T. Smith, R. Joshi, and C. Costa, “Serving GLM-5.2 for agentic workloads on llm-d,” llm-d blog, Jul. 2026.
- [28] M. Al-Fares *et al.*, “Hedera: dynamic flow scheduling for data center networks,” in *Proc. NSDI*, 2010.
- [29] M. Alizadeh *et al.*, “CONGA: distributed congestion-aware load balancing for datacenters,” in *Proc. ACM SIGCOMM*, 2014.
- [30] Y. Zhu *et al.*, “Congestion control for large-scale RDMA deployments,” in *Proc. ACM SIGCOMM*, 2015.
- [31] N. Farrington *et al.*, “Helios: a hybrid electrical/optical switch architecture for modular data centers,” in *Proc. ACM SIGCOMM*, 2010.
- [32] G. Wang *et al.*, “c-Through: part-time optics in data centers,” in *Proc. ACM SIGCOMM*, 2010.
- [33] H. Liu *et al.*, “Lightwave fabrics: at-scale optical circuit switching for datacenter and machine learning systems,” in *Proc. ACM SIGCOMM*, 2023.
- [34] L. Poutievski *et al.*, “Jupiter evolving: transforming Google’s datacenter network via optical circuit switches and software-defined networking,” in *Proc. ACM SIGCOMM*, 2022.
- [35] R. Urata *et al.*, “Mission Apollo: landing optical circuit switching at datacenter scale,” arXiv:2208.10041, 2022.
- [36] T. J. Seok, N. Quack, S. Han, R. S. Muller, and M. C. Wu, “Large-scale broadband digital silicon photonic switches with vertical adiabatic couplers,” *Optica*, vol. 3, no. 1, pp. 64–70, 2016.
- [37] M. Wade *et al.*, “An error-free 1 Tbps WDM optical I/O chiplet and multi-wavelength multi-port laser,” in *Proc. OFC*, 2021, paper F3C.6.
- [38] Optical Interworking Forum, “Implementation agreement for a 3.2 Tb/s co-packaged (CPO) module,” OIF-Co-Packaging-3.2T-Module-01.0, 2023.
- [39] CALIENT Technologies, “CALIENT’s optical circuit switch,” S320 data sheet, 2022.
- [40] FS, “400G DR4 OSFP PAM4 1310nm 500m MTP/MPO SMF silicon photonics transceiver,” data sheet, 2022.
- [41] Sumitomo Electric, “Single mode CWDM4 mux/demux coupler, FlexBeamGuide-WD,” product page, accessed Oct. 2026.
- [42] *IEEE Standard for Ethernet*, IEEE Std 802.3-2022, Clause 124 (400GBASE-DR4), and IEEE Std 802.3cu-2021, Clause 151 (400GBASE-FR4).
- [43] 100G Lambda MSA, “400G-FR4 technical specification,” Rev. 2.0, 2018.
- [44] *Optical Fiber Cabling and Components Standard*, ANSI/TIA-568.3-D, 2016.
- [45] Microsoft, “Azure LLM inference trace 2024,” AzurePublicDataset, GitHub, 2024.
- [46] J. Stojkovic, C. Zhang, Í. Goiri, J. Torrellas, and E. Choukse, “DynamoLLM: designing LLM inference clusters for performance and energy efficiency,” in *Proc. IEEE HPCA*, 2025.
- [47] M. Alizadeh *et al.*, “pFabric: minimal near-optimal datacenter transport,” in *Proc. ACM SIGCOMM*, 2013.
- [48] B. Montazeri, Y. Li, M. Alizadeh, and J. Ousterhout, “Homa: a receiver-driven low-latency transport protocol using network priorities,” in *Proc. ACM SIGCOMM*, 2018.
- [49] M. Khani *et al.*, “SiP-ML: high-bandwidth optical network interconnects for machine learning training,” in *Proc. ACM SIGCOMM*, 2021.
- [50] W. M. Mellette *et al.*, “RotorNet: a scalable, low-complexity, optical datacenter network,” in *Proc. ACM SIGCOMM*, 2017.
- [51] H. Ballani *et al.*, “Sirius: a flat datacenter network with nanosecond optical switching,” in *Proc. ACM SIGCOMM*, 2020.
- [52] W. Wu, X. Mu, X. Chen, B. Tang, X. Cui, Y. Dou, and Z. Zhu, “xSwitch: an adaptive O-E-integrated interconnect for scale-up networks,” in *Proc. ACM SIGCOMM Workshop on Hot Topics in Optical Technologies and Applications in Networking (HotOptics)*, 2026, pp. 2116–2121.
- [53] J. Ding, Y. Nishant, C. Ambati, J. Kamati, and T. Diep, “A photonic-CXL memory appliance for scalable KV cache management in LLM inference,” arXiv:2607.27187, 2026.
- [54] Y. Liu *et al.*, “CacheGen: KV cache compression and streaming for fast large language model serving,” in *Proc. ACM SIGCOMM*, 2024.
- [55] Z. Liu *et al.*, “KIVI: a tuning-free asymmetric 2bit quantization for KV cache,” in *Proc. ICML*, 2024.
- [56] Z. Liu *et al.*, “KVServe: service-aware KV cache compression for communication-efficient disaggregated LLM serving,” in *Proc. ACM SIGCOMM*, 2026.