POSTER: Beyond Probabilistic Data Structures for AI/ML Workload Monitoring

Davide Palmiotti
Politecnico Di Milano

Michele Ferrero
Huawei Technologies, France

Gabriele Castellano
Huawei Technologies, France

Massimo Gallo
Huawei Technologies, France

Gianni Antichi
Politecnico Di Milano

1 INTRODUCTION

The widespread adoption of large language models (LLMs) in various domains has led to the development of increasingly complex architectures, often exceeding trillions of parameters [5]. Training these models requires a massive computational infrastructure, relying on datacenters that deploy thousands of GPUs operating in parallel [5, 7]. In these datacenters, maintaining flow and packet-level statistics within ASIC switches is a critical telemetry strategy for performance analysis and debugging.

However, the scalability of existing telemetry solutions is increasingly constrained by the rapid growth of GPU training clusters, whose scale is expanding faster than the memory capacity of modern switch data planes. This memory bottleneck is further exacerbated by the shift in large-scale GPU communication toward aggressive multipath strategies, such as packet spraying [1, 2]. Unlike traditional Equal-Cost Multi-Path (ECMP) routing, these techniques significantly increase the volume of concurrent flow states that switches need to track. Consequently, while classical telemetry data structures are efficient for traditional Data Center Networks (DCN), they are suboptimal for the distinct traffic characteristics and scale of modern AI/ML clusters.

Figure 1 shows the in-switch memory utilization required to track 99% of flows using three representative data structures for the counting of per-flow packets, namely CountMinSketch, ElasticSketch [9], and FlowLidar [6]. Flows were derived from three synthesized traces that replicate the traffic profile of a ToR switch within a 2048-GPUs fat-tree topology running large-scale AI/ML training job such as DeepSeek-v3, DeepSeek-v4, and projected future models. These traces were generated by applying the traffic characteristics i.e., number of flows and the flow-size distribution, observed in SimAI [8]. Accurately tracking at least 99% of flows is critical for distributed ML training workloads due to the tight synchronization required across thousands of GPUs. Even a single delayed or degraded flow can introduce stragglers that stall collective communication operations and slow down the entire training job. Consequently, trading telemetry accuracy to lower memory consumption is undesirable in this

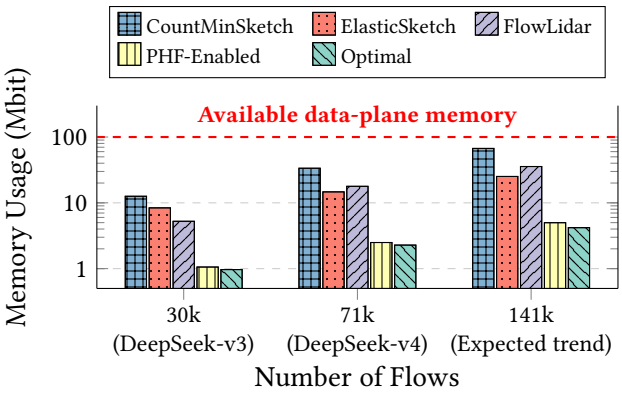

Figure 1: Memory footprint comparison of state-of-the-art telemetry solutions for different scenarios. Modern switch ASICs provide O(10MB) of data-plane memory.

environment, as missing even a small fraction of anomalous flows can severely impact end-to-end performance.

Contextualizing these requirements against the available memory for stateful operations in commodity programmable switches [3], reveals a critical bottleneck. As each new generation of AI models grows in scale and complexity, the resulting explosion in concurrent network flows causes telemetry data structures to rapidly exhaust the limited on-chip memory available within switch ASICs. This creates a critical resource contention problem: memory heavily allocated to monitoring and flow accounting is no longer available for other essential data-plane functions, such as access control lists (ACLs), routing and forwarding tables, to name a few. Most importantly, existing telemetry solutions often operate far from the theoretical memory lower bound, typically requiring between **5×** and **13×** more memory than the ideal baseline of one counter per observed flow, placing high pressure on already scarce switch resources.

This divergence stems from the assumption that *traffic is unknown apriori* in traditional DCN. To prevent ASIC memory exhaustion when monitoring a potentially unbounded

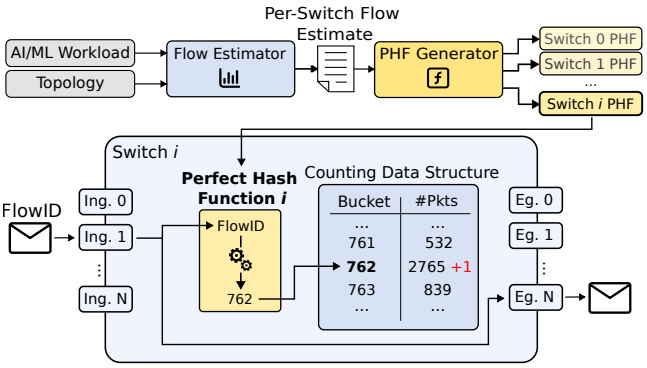

Figure 2: Overview of an in-switch per-flow packet counting solution using perfect hashing.

flow space, traditional telemetry solutions use probabilistic data structures. By compressing flow states via hashing, these designs intentionally trade accuracy for a smaller memory footprint. However, because standard hash functions are prone to collisions, these systems are forced to heavily over-provision memory to maintain very high accuracy i.e., 99-100%, required by modern applications, ultimately driving the severe overhead observed in Figure 1.

A fundamental distinction of AI/ML training workloads is that *flows are highly predictable*; indeed, GPUs consistently communicate using repetitive patterns across training iterations. This determinism allows us to improve upon existing solutions by leveraging *perfect hashing* [4]. A perfect hash function (PHF), denoted as $h(k)$, maps every key in a known set of flows S to a unique, distinct index. For in-switch packet counting, a PHF guarantees that each flow maps to a discrete, unique memory location, thereby eliminating collisions and the resulting need for memory overprovisioning. While constructing a PHF requires prior knowledge of the flow set S, an infeasible requirement for the stochastic traffic of traditional datacenters, the deterministic nature of AI/ML workloads makes this approach highly practical in our settings. As shown in Figure 1, a PHF-based architecture enables the tracking of all concurrent flows with negligible memory overhead, representing a substantial improvement over traditional designs. Furthermore, the total elimination of collisions ensures 100% measurement accuracy, achieving deterministic precision that is unattainable with probabilistic data structures.

2 OUR PROPOSAL

Figure 2 details our perfect hashing solution for per-flow packet counting in switches. Prior to training, a *flow estimator* ingests the AI/ML workload and datacenter routing topology to predict the exact flow set traversing each switch, using either simulation or analytical modeling. A *PHF Generator* then uses these per-switch estimates to construct custom PHFs, which are subsequently deployed to the corresponding switches. At runtime, upon packet reception, the switch extracts a flow identifier (FlowID) e.g., source IP address, or full 5-tuple, matching the granularity of the installed PHF. The FlowID is then hashed through the PHF, which will output a unique memory location or *bucket* holding the corresponding packet counter for that specific flow. The switch updates this counter while simultaneously forwarding the packet to its next hop. By eliminating hash collisions entirely, our approach bounds the required memory to exactly one bucket per expected flow, drastically outperforming classical data structures. Furthermore, the memory overhead required to store modern PHFs is negligible (about 3 bits/flow [4]).

While this architecture offers significant potential, we acknowledge that it introduces some critical challenges necessitating further in-depth research and analysis.

Challenge 1: In-hardware implementation. PHF storage within switch memory is minimal, as modern PHFs require about 3 bits/key. However, the resources required to compute the hash in a switch with a PHF need careful consideration. Modern PHF are more computationally advanced than traditional hash functions. This complexity could pose a challenge for implementing these functions in the switch ASIC. Alternatively, a cross-layer approach could be leveraged where end-hosts pre-calculate the PHF and tag packets with the target memory index, offloading the computational burden from the switch fabric to the network edge. For example, it could prove useful to leverage the IPv6 *flow label* field within IPv6 header to implement the proposed solution.

Challenge 2: Estimation Drift. A PHF only guarantees collision-free mapping for its predefined input set. In a dynamic datacenter environment, actual ground-truth traffic may occasionally deviate from the initial estimate, necessitating a system that is either aware of these discrepancies or resilient to them. For example, this is the case of unpredictable flows resulting from unexpected network failures. Any of these cases may result in an unexpected flow identifier to be hashed through the PHF, which may map to an occupied bucket and cause a collision. Beyond refining the per-flow estimate to account for anomalous behavior, system robustness can be enhanced by incorporating an auxiliary validation mechanism. This component identifies out-of-set flows and redirects them to a secondary fallback structure, ensuring accurate packet counting for traffic not captured by the primary data structure.

REFERENCES

[1] Tommaso Bonato, Abdul Kabbani, Ahmad Ghalayini, Michael Papamichael, Mohammad Dohadwala, Lukas Gianinazzi, Mikhail Khalilov, Elias Achermann, Daniele De Sensi, and Torsten Hoefler. 2026. REPS:

Recycled Entropy Packet Spraying for Adaptive Load Balancing and Failure Mitigation. In *European Conference on Computer Systems (EuroSys)*. ACM.

[2] Torsten Hoefler, Karen Schramm, Eric Spada, Keith Underwood, Cedell Alexander, Bob Alverson, Paul Bottorff, Adrian Caulfield, Mark Handley, Cathy Huang, Costin Raiciu, Abdul Kabbani, Eugene Opsasnick, Rong Pan, Adee Ran, and Rip Sohan. 2025. Ultra Ethernet's Design Principles and Architectural Innovations. *arXiv preprint*.

[3] Daehyeok Kim, Zaoxing Liu, Yibo Zhu, Changhoon Kim, Jeongkeun Lee, Vyas Sekar, and Srinivasan Seshan. 2020. TEA: Enabling State-Intensive Network Functions on Programmable Switches. In *Special Interest Group on Data Communication (SIGCOMM)*. ACM.

[4] Hans-Peter Lehmann, Thomas Mueller, Rasmus Pagh, Giulio Ermanno Pibiri, Peter Sanders, Sebastiano Vigna, and Stefan Walzer. 2026. Modern Minimal Perfect Hashing: A Survey. *ACM Computing Surveys, Volume 58, Number: 10*.

[5] Qingkai Meng, Hao Zheng, Zhenhui Zhang, ChonLam Lao, Chengyuan Huang, Baojia Li, Ziyuan Zhu, Hao Lu, Weizhen Dang, Zitong Lin, Weifeng Zhang, Lingfeng Liu, Yuanyuan Gong, Chunzhi He, Xiaoyuan Hu, Yinben Xia, Xiang Li, Zekun He, Yachen Wang, Xianneng Zou, Kun Yang, Gianni Antichi, Guihai Chen, and Chen Tian. 2025. Astral: A datacenter infrastructure for large language model training at scale. In *Special Interest Group on Data Communication (SIGCOMM)*. ACM.

[6] Andrea Monterubbiano, Jonatan Langlet, Stefan Walzer, Gianni Antichi, Pedro Reviriego, and Salvatore Pontarelli. 2023. Lightweight Acquisition and Ranging of Flows in the Data Plane. *Measurement and Analysis of Computing Systems, Volume: 7, Number: 3*.

[7] Kun Qian, Yongqing Xi, Jiamin Cao, Jiaqi Gao, Yichi Xu, Yu Guan, Binzhang Fu, Xuemei Shi, Fangbo Zhu, Rui Miao, Chao Wang, Peng Wang, Pengcheng Zhang, Xianlong Zeng, Eddie Ruan, Zhiping Yao, Ennan Zhai, and Dennis Cai. 2024. Alibaba HPN: A Data Center Network for Large Language Model Training. In *Special Interest Group on Data Communication (SIGCOMM)*. ACM.

[8] Xizheng Wang, Qingxu Li, Yichi Xu, Gang Lu, Dan Li, Li Chen, Heyang Zhou, Linkang Zheng, Sen Zhang, Yikai Zhu, Yang Liu, Pengcheng Zhang, Kun Qian, Kunling He, Jiaqi Gao, Ennan Zhai, Dennis Cai, and Binzhang Fu. 2025. SimAI: Unifying Architecture Design and Performance Tuning for Large-Scale Large Language Model Training with Scalability and Precision. In *Networked Systems Design and Implementation (NSDI)*. USENIX Association.

[9] Tong Yang, Jie Jiang, Peng Liu, Qun Huang, Junzhi Gong, Yang Zhou, Rui Miao, Xiaoming Li, and Steve Uhlig. 2018. Elastic sketch: adaptive and fast network-wide measurements. In *Special Interest Group on Data Communication (SIGCOMM)*. ACM.