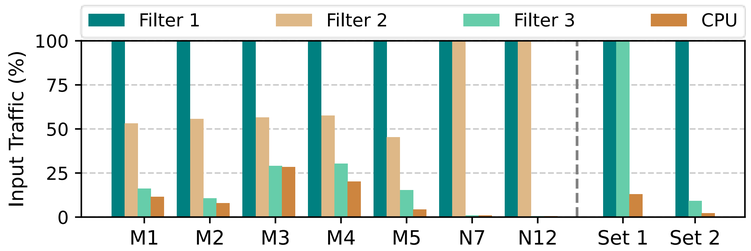
RapidQ: Lightweight Queueing Abstraction for Rapid Simulation and Automated Tuning of Input-Dependent Streaming Pipelines

Shashank Obla, Carnegie Mellon University | Bin Li, Intel Corporation | James C. Hoe, Carnegie Mellon University

Carnegie Mellon University

Given a high-performance parametrized design, what should the module throughputs and buffer sizes be for resource efficiency?

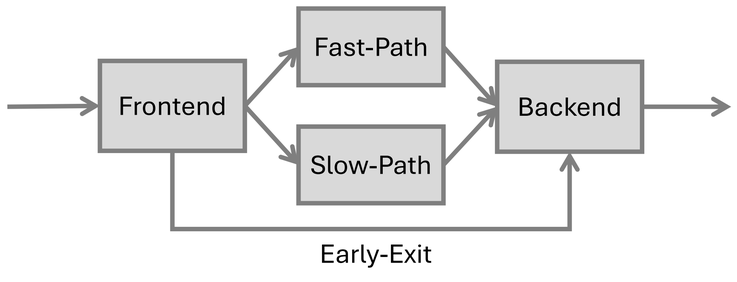
- Traces dictate performance for input-dependent designs
- Near cycle-accurate simulation for large real-world workloads, limits the iterative tuning agility



Higher level abstractions are essential to enabling rapid per-workload retuning

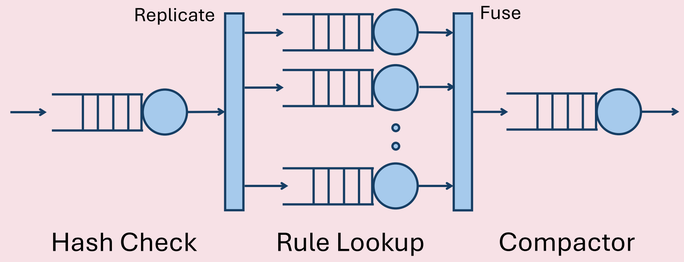
INSIGHT: Streaming pipelines capture input-dependence in the graph structure

- Input-dependence does not map well to the spatial parallelism easily exploitable using HLS
- Streaming paradigm allows decoupling of parallelizable compute from data-dependent infrastructure kernels
- Each kernel can be abstracted to its unroll factor, and the graph as such kernels connected by buffers



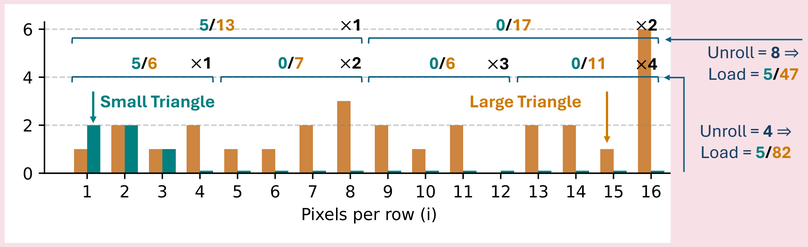
RapidQ Modeling and Automated Tuning Flow

Step 1: Streaming Graph is captured as a Queuing network

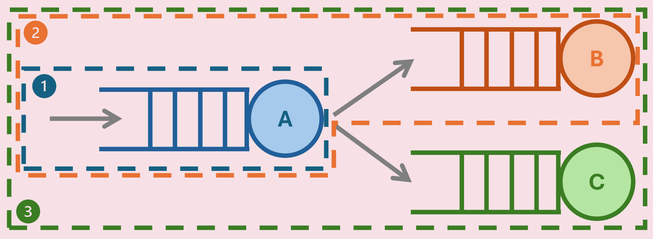


Step 2: Workload is traced through one-shot C Simulation

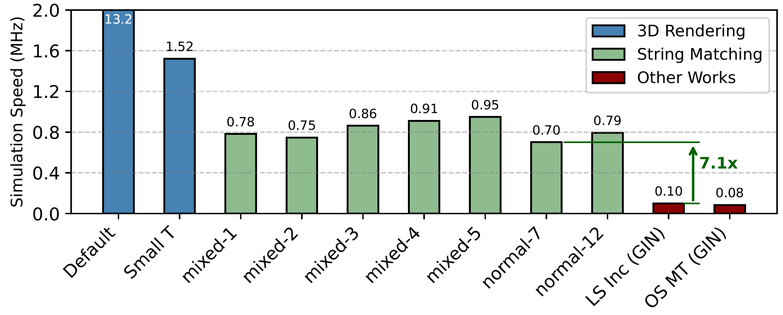
- Load per packet is stored at server/queue boundaries denoting buffer size or # of elements to be processed
- Input-dependent sparsity is encoded as histograms



Step 3: Automated Tuning breaks down the design space with the RapidQ Simulator providing performance estimates

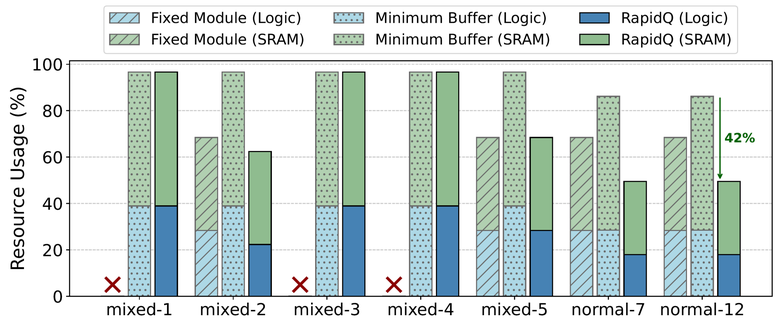


RapidQ trades off simulation accuracy for speed for rapid tuning turnaround time



RapidQ is 7x faster than state-of-the-art[1]

Fast Simulation enables RapidQ to co-tune buffer depths and kernel parallelism



RapidQ saves over 40% resources in a real-world network security workload[2]

[1]R. Sarkar and C. Hao, "Lightningsim: Fast and accurate trace-based simulation for high-level synthesis," in 2023 IEEE 31st Annual International Symposium on Field-Programmable Custom Computing Machines

[2]Z. Zhao, H. Sadok, N. Atre, J. C. Hoe, V. Sekar,and J. Sherry, "Achieving 100gbps intrusion prevention on a single server," in 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)

[3]Z. Zhao, J. Melber, S. Sahay, S. Obla, E. Nurvitadhi, and J. C. Hoe, "Exploiting the common case when accelerating input-dependent stream processing by fpga," IEEE Transactions on Computers, vol. 72, no. 5, pp. 1343–1355, 2023.