Lightweight Queueing Abstraction for Rapid Simulation and Automated Tuning of Input-Dependent Streaming Pipelines on FPGAs

Shashank Obla
Carnegie Mellon University
Pittsburgh, PA, USA
sobla@andrew.cmu.edu

Bin Li
Intel Corporation
Hillsboro, OR, USA
bin.li@intel.com

James C. Hoe
Carnegie Mellon University
Pittsburgh, PA, USA
jhoe@andrew.cmu.edu

***Abstract*—The programmability of FPGAs enables designs to be tuned to specific deployment and use-cases. This capability is critical for input-dependent streaming pipelines, whose optimal configuration varies not only across resource limits and performance targets, but also with the data being processed. However, existing methods fail to scale for large, real-world designs with long workloads. To address this challenge, we propose RapidQ, a performance model that allows designers to capture the dataflow of their streaming pipeline as a queueing system, enabling fast case-by-case re-tuning for resource efficiency. Unburdened by the functional details of the design, a trace model drives a fast queueing simulation that can predict the performance of the pipeline across various buffer sizes and module throughput configurations without repeating full functional simulations. Our simulator is over 7x faster than the state-of-the-art and yields up to 42% resource savings for real-world workloads.**

Motivation. Existing DSE methodologies face significant limitations when applied to input-dependent systems. While many FPGA DSE tools utilize static analysis from HLS tools to accurately predict performance, they fail to account for input-driven variations. Runtime execution-based approaches using incremental compilation [1] can achieve the high accuracy and observability needed to optimize module throughputs. Conversely, tuning buffer sizes exposes a much broader design space, where schedule-augmented software simulation [2] provides the necessary agility. However, both approaches depend on the repeated execution of functionality in the design to model input-dependent behavior. Furthermore, tuning buffer sizes in isolation while modules remain under-provisioned can result in excessive memory usage and a resource-inefficient design; tuning only modules can be equally inefficient.

RapidQ Performance Model. In this paper, we present RapidQ, a systematic approach to modeling and tuning input-dependent streaming pipelines for FPGAs. The module boundaries in a streaming pipeline are deliberately constructed to manage the design complexity and performance of input-dependent applications. Leveraging this structure, RapidQ models the pipeline as a queueing system, where buffers are represented as queues and processing modules as servers, abstracting the functionality within. We employ functional software simulation, once per trace, to quantify the load each input data element generates for every server and the storage space required in the queues.

RapidQ Simulator. To predict end-to-end performance, we implement a fast queueing simulator driven by the RapidQ model. By combining HLS synthesis reports with designer annotations, the simulator programmatically converts the functional simulation trace into precise timing characteristics for each data element for any server throughput configuration. This eliminates the need for repeated functional simulations, significantly reducing the turnaround time for design space exploration.

RapidQ Auto-Tuning. Lastly, driven by the queueing performance simulation, we develop an automated tuning flow to identify resource-efficient configurations for input-dependent streaming pipelines that meet specific throughput targets for a given input workload. The exploration space encompasses both buffer sizes and parallelism/unroll parameters for the streaming modules. We formulate a resource cost function that normalizes heterogeneous FPGA resources into a unified percentage area metric. This enables the tuning loop to efficiently tradeoff module resources against buffering capacity, minimizing the total resource footprint of the pipeline.

Evaluation. We evaluate our modeling framework and tuning flow using a 3D Rendering pipeline, and a Multi-String Matching Pipeline, evaluated against real-world traces. RapidQ accurately predicts the performance of the pipelines across diverse module throughput configurations (within 3% error) while reducing simulation time by more than **7x** compared to state-of-the-art HLS simulators [2]. Furthermore, using our tuning flow, we show that co-optimizing module throughputs and buffer sizes saves up to **42%** in resource footprint compared to baselines that tune these parameters in isolation.

References

[1] D. Park and A. DeHon, "Refine: Runtime execution feedback for incremental evolution on fpga designs," in *Proceedings of the 2024 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 108–118. [Online]. Available: https://doi.org/10.1145/3626202.3637560

[2] 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 (FCCM)*, May 2023, p. 1–11. [Online]. Available: https://ieeexplore.ieee.org/document/10171498/